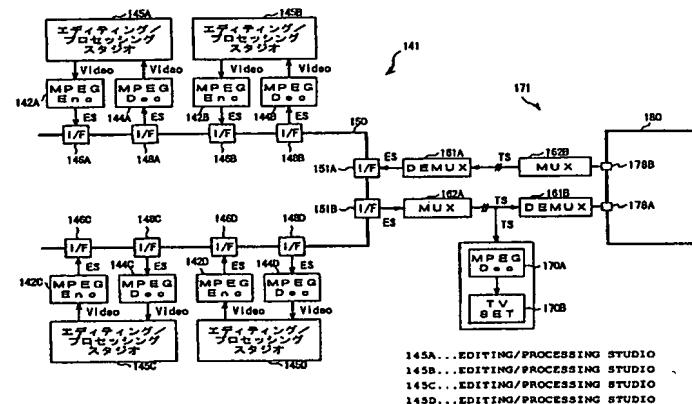




(51) 国際特許分類7 H04N 7/08, 7/24	A1	(11) 国際公開番号 WO00/46989
		(43) 国際公開日 2000年8月10日(10.08.00)
(21) 国際出願番号 PCT/JP00/00628		(81) 指定国 BR, CN, JP, KR, US, 欧州特許 (DE, FR, GB, IT, NL)
(22) 国際出願日 2000年2月4日(04.02.00)		添付公開書類 国際調査報告書
(30) 優先権データ 特願平11/29378 1999年2月5日(05.02.99)	JP	
(71) 出願人 (米国を除くすべての指定国について) ソニー株式会社(SONY CORPORATION)[JP/JP] 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo, (JP)		
(72) 発明者 ; および (75) 発明者 / 出願人 (米国についてのみ) 田原勝己(TAHARA, Katsumi)[JP/JP] 村上芳弘(MURAKAMI, Yoshihiro)[JP/JP] 〒141-0001 東京都品川区北品川6丁目7番35号 ソニー株式会社内 Tokyo, (JP)		
(74) 代理人 弁理士 田辺恵基(TANABE, Shigemoto) 〒150-0001 東京都渋谷区神宮前1丁目11番11-508号 グリーンファンタジアビル5階 Tokyo, (JP)		

(54) Title: ENCODING DEVICE, ENCODING METHOD, DECODING DEVICE, DECODING METHOD, CODING SYSTEM AND CODING METHOD

(54) 発明の名称 エンコーディング装置、エンコーディング方法、デコーディング装置、デコーディング方法、コーディングシステム、コーディング方法



(57) Abstract

The invention relates to an encoding system for encoding input video data and to a decoding system for decoding encoded streams. Specifically, an encoded stream is accompanied in transmission so that ancillary data and information characteristic of the video data may not be lost after a number of repetitions of MPEG encoding and MPEG decoding. An MPEG encoder extracts the ancillary data attached to input video data, and transmits the extracted ancillary data together with the encoded stream by inserting the ancillary data as Ancillary_data() in the encoded stream. An MPEG decoder extracts the ancillary data from the encoded stream, and attaches the extracted ancillary data to baseband video data generated by MPEG decoding.

(57)要約

本発明は、入力ビデオデータを符号化するためエンコーディングシステム及び符号化ストリームをデコードするためのデコーディングシステムに関する。具体的には、MPEGエンコーディング処理及びMPEGデコーディング処理を繰り返したとしても、ビデオデータに付加されているアンシラリーデータやビデオデータ固有の情報が無くならないように、符号化ストリームと一緒に伝送するためのシステム及び方法を提供するものである。

MPEGエンコーダは、その入力ビデオデータに付加されているアンシラリーデータを抽出し、抽出したアンシラリーデータを符号化ストリーム中にAn-
cillary_data()として挿入することによって、符号化ストリームと一緒にこのアンシラリーデータを伝送する。MPEGデコーダは、符号化ストリームからアンシラリーデータを抽出し、MPEGデコーディング処理によって生成されたベースバンドビデオデータに抽出したアンシラリーデータを付加する

。

PCTに基づいて公開される国際出願のパンフレット第一頁に掲載されたPCT加盟国を同定するために使用されるコード(参考情報)

AE アラブ首長国連邦	DM ドミニカ	KZ カザフスタン	RU ロシア
AG アンティグア・バーブーダ	DZ アルジェリア	LC セントルシア	SD スーダン
AL アルバニア	EE エストニア	LI リヒテンシュタイン	SE スウェーデン
AM アルメニア	ES スペイン	LK スリ・ランカ	SG シンガポール
AT オーストリア	FI フィンランド	LR リベリア	SI スロヴェニア
AU オーストラリア	FR フランス	LS レソト	SK スロ伐キア
AZ アゼルバイジャン	GA ガボン	LT リトアニア	SL シエラ・レオネ
BA ボスニア・ヘルツェゴビナ	GB 英国	LU ルクセンブルグ	SN セネガル
BB バルバドス	GD グレナダ	LV ラトヴィア	SZ スワジランド
BE ベルギー	GE グルジア	MA モロッコ	TD チャード
BF ブルガリア・ファン	GH ガーナ	MC モナコ	TG トーゴー
BG ブルガリア	GM ガンビア	MD モルドヴァ	TJ タジキスタン
BJ ベナン	GN ギニア	MG マダガスカル	TM トルクメニスタン
BR ブラジル	GR ギリシャ	MK マケドニア旧ユーゴスラヴィア	TR トルコ
BY ベラルーシ	GW ギニア・ビサオ	共和国	TT トリニダッド・トバゴ
CA カナダ	HR クロアチア	ML マリ	TZ タンザニア
CF 中央アフリカ	HU ハンガリー	MN モンゴル	UA ウクライナ
CG コンゴ	ID インドネシア	MR モーリタニア	UG ウガンダ
CH スイス	IE アイルランド	MW マラウイ	US 米国
CI コートジボアール	IL イスラエル	MX メキシコ	UZ ウズベキスタン
CM カメルーン	IN インド	MZ モザンビーク	VN ベトナム
CN 中国	IS アイスランド	NE ニジエール	YU ユーロースラヴィア
CR コスタ・リカ	IT イタリア	NL オランダ	ZA 南アフリカ共和国
CU キューバ	JP 日本	NO ノールウェー	ZW ジンバブエ
CY キプロス	KE ケニア	NZ ニュージーランド	
CZ チェコ	KG キルギスタン	PL ポーランド	
DE ドイツ	KP 北朝鮮	PT ポルトガル	
DK デンマーク	KR 韓国	RO ルーマニア	

明細書

エンコーディング装置、エンコーディング方法、デコーディング装置、デコーディング方法、コーディングシステム、コーディング方法

技術分野

本発明は入力ビデオデータを符号化するためエンコーディングシステム及び符号化ストリームをデコードするためのデコーディングシステムに関する。

背景技術

近年、テレビジョンプログラムを制作及び放送する放送局においては、ビデオデータを圧縮/符号化するために、ISO/IEC 13818 として規定されているMPEG (Moving Picture Experts Group) 技術が一般的に使われるようになってきた。特に、ビデオカメラなどによって生成されたビデオデータを、テープやディスクなどのランダムアクセス可能な記録媒体素材に記録する場合や、放送局内において制作されたビデオプログラムを、ケーブルや衛星を介して伝送する場合には、このMPEGがデファクトスタンダードになりつつある。

このMPEG技術は、ピクチャ間の予測符号化方式を用いることによって、圧縮効率を向上させることができる符号化技術である。具体的には、MPEG規格においては、フレーム内予測やフレーム予測を組み合わせた複数の予測符号化を採用しており、その予測方式に従って、各ピクチャはIピクチャ (Intra Picture)、Pピクチャ (Predictive Picture)、またはBピクチャ (Bidirectionally predictive Picture) のいずれかのピクチャタイプを使用してエンコード処理が行われる。Iピクチャは、他のピクチャからの予測が行われず、フレーム内符号化処理されたピクチャであり、Pピクチャは、時間的に前(過

去) の I ピクチャ又は P ピクチャからフレーム間順方向予測符号化されたピクチャであり、B ピクチャは、時間的に前(過去)の I ピクチャ又は P ピクチャと、時間的に後(未来)の I ピクチャ又は P ピクチャとの両方向のピクチャから双方向予測符号化されたピクチャである。

図 1 は、放送局内及放送局間におけるビデオプロセッシングシステムの一例を示している。前述したように、放送局内や放送局間では、送信側システムとして設けられた第 1 のビデオプロセッサ 1 から、受信側システムとしてもうけられた第 2 のビデオプロセッサ 4 へ、ソースビデオデータを伝送するために、送信システムと受信システムとの間において、図 1 のようにM P E G エンコーダ及びM P E G デコーダを使用することが提案されている。

第 1 のビデオプロセッサ 1 は、D 1 フォーマットのコンポーネント形式のベースバンドビデオ等のソースビデオデータを受け取り、そのソースビデオデータに対して、編集処理、特殊効果処理、合成処理などを行うための装置である。さらには、このビデオプロセッサ 1 は、クローズドキャプションやテレテキストデータ等の補助データ(アンシラリーデータ)を受け取り、そのアンシラリーデータをソースビデオデータのプランкиング期間に付加する処理も行う。従って、ビデオプロセッサ 1 から出力されるビデオデータのプランкиング期間には、アンシラリーデータが埋め込まれて(エンベデッド)されている。

M P E G エンコーダ 2 は、ビデオプロセッサ 1 からビデオデータを受け取り、そのビデオデータをエンコードすることによって符号化ストリームを生成する。この符号化ストリームは、エレメンタリーストリームとも称される。誰もが知っているように、テレビジョン信号というのは、アクティブビデオエリアと呼ばれる実際のビデオデータエリアの上下及び左右に垂直プランкиング期間及び水平プランкиング期間が設けられており、上述したアンシラリーデータは、このプランкиング区間に挿入されている。

しかしながら、M P E G 規格では、入力ビデオデータの中において、実際に画素が存在するアクティブビデオエリアのみを符号化しなければいけないと規定さ

れている。つまり、符号化ストリームは、入力ビデオデータのアクティブエリアのみを符号化処理したデータであって、その符号化ストリームにはプランкиング期間に重複されていたアンシラリーデータは含まれていないのである。言いかえると、MPEGエンコーダ2によって入力ビデオデータの符号化処理を行うことによって、入力ビデオデータに重複されていたアンシラリーデータが無くなってしまうということである。

MPEGデコーダ3は、MPEGエンコーダから符号化ストリームを受け取り、この符号化ストリームをデコードして復号化ビデオデータを生成する。復号化ビデオデータは、第2のビデオプロセッサ4に供給される。このMPEGデコーダ3に供給された符号化ストリーム中には、アンシラリーデータに関する情報が含まれていないので、当然ながら、この復号化ビデオデータにもアンシラリーデータに関する情報が含まれていない。

つまり、送出側システムから受信側システムにビデオデータを伝送する途中において、MPEGエンコード処理及びMPEGデコード処理を行った場合には、送出側である第1のビデオプロセッサ1から受信側である第2のビデオプロセッサ4には、アクティブエリアに相当するビデオデータは伝送することはできるが、第1のビデオプロセッサにおいてビデオデータのプランкиング期間に付加したアンシラリーデータは伝送されなという問題が発生する。

さらに言うと、送出側システムから受信側システムにビデオデータを伝送する途中において、MPEGエンコード処理及びMPEGデコード処理を行った場合には、アクティブエリアのビデオデータしか伝送されないということは、ソースビデオデータが有している固有の情報も受信側システムに伝送されないという問題もある。このソースビデオデータに係る固有の情報とは、ソースビデオデータそのものが持っている情報であって、例えば、全画素エリアに対するプランкиングエリアの位置又はアクティブビデオエリアの位置などである。具体的には、ソースビデオデータの全画素エリアの垂直方向において何ライン目からアクティブビデオのラインが始まり、全画素エリアの水平方向において何ピクセル目からアク

タイプビデオエリアが始まるのかを示す情報である。

次に、図2を参照して、3-2プルダウン処理が行われたビデオデータに関する処理について説明する。図2は、24Hzのフレーム周波数を有するビデオデータと30Hzのフレーム周波数を有するビデオデータの両方のビデオデータに関する処理を行う場合の、放送局内及び又は放送局間におけるビデオプロセッシングシステムの一例を示している。

3-2プルダウン処理回路5は、24Hz（毎秒24コマ）のフレームレートのビデオデータを受け取って、30Hz（毎秒30コマ）のフレームレートのビデオデータを生成するための回路である。映画等において使用されているフィルム素材は、劇場用として、24Hz（毎秒24コマ）のフレームレートで光学フィルム上に記録された素材であって、29.97Hzのフレームレートを有するNTSC方式のテレビジョン信号とはフレームレートが全く異なっている。従つて、フィルム素材をテレビジョン信号に変換するために、24コマから30フレームを生成する処理が行われている。

図3A及び図3Bを参照して、この3-2プルダウン処理について説明する。

図3Aは、24Hzのフレームレートのソースビデオデータを表し、図3Bは、3-2プルダウン処理によって30Hzのフレームレートに変換されたビデオデータを表している。図3A及び図3Bに示すように、3-2プルダウン処理においては、フィールドF1のトップフィールドt1をリピートすることによってリピートフィールドt1'を作成し、フレームF3のボトムフィールドb3をリピートすることによってリピートフィールドb3'を生成する。つまり、3-2プルダウン処理は、2個のフィールドを、所定のシーケンスで3個のフィールドに変換することによって、24Hzのビデオデータを30Hzのビデオデータに変換する処理なのである。

第1のビデオプロセッサ1は、図1において説明したように、30Hzのソースビデオデータを受け取り、そのソースビデオデータに対して、編集処理、特殊効果処理、合成処理などを行うための装置である。さらには、このビデオプロセ

ッサ1は、クローズドキャプションやテレテキストデータ等の補助データ（アンシラリーデータ）を受け取り、そのアンシラリーデータをソースビデオデータのプランкиング期間に付加する処理も行う。アンシラリーデータを付加する処理は、30Hzのフレーム周波数を有するビデオデータに対して行われる処理であるので、30Hzのフレーム周波数を有するビデオデータに含まれる全フィールドに対して、アンシラリーデータが付加される。つまり、トップフィールドt1、t2、……及びボトムフィールドb1、b2、……だけに、アンシラリーデータが付加されるのではなくて、リピートフィールドt1'及びリピートフィールドb3'にもアンシラリーデータが付加されるのである。

逆3-2プルダウン処理回路6は、上述した3-2プルダウン処理により得られた30Hzのフレームレートのビデオデータを受け取って、24Hzのフレームレートのビデオデータを生成するための回路である。具体的には、逆3-2プルダウン処理回路7は、図3Cに示すように、逆3-2プルダウン処理によって挿入されたリピートフィールドt1'及びb3'を除去するための回路である。この、逆3-2プルダウン処理回路7は、MPEGエンコード処理を行う前に行われなければならない処理である。なぜなら、このリピートフィールドは、3-2プルダウン処理によって挿入された冗長なフィールドであって、削除したとしても何ら画質劣化が発生しないからである。

MPEGエンコーダ2は、図1において説明したMPEGエンコーダ2と同じであって、逆プルダウン処理回路6から24Hzのビデオデータを受け取り、そのビデオデータをエンコードすることによって符号化ストリームを生成する。

しかしながら、MPEG規格では、入力ビデオデータの中において、実際に画素が存在するアクティブビデオエリアのみを符号化しなければいけないと規定されている。つまり、符号化ストリームは、入力ビデオデータのアクティブエリアのみを符号化処理したデータであって、その符号化ストリームにはプランкиング期間に重畠されていたアンシラリーデータは含まれていないのである。言いかえると、MPEGエンコーダ2によって入力ビデオデータの符号化処理を行うことに

よって、入力ビデオデータに重畠されていたアンシラリーデータが無くなってしまうということである。

MPEGデコーダ3は、図1において説明したMPEGデコーダ3と同じであって、MPEGエンコーダから符号化ストリームを受け取り、この符号化ストリームをデコードして復号化ビデオデータを生成する。尚、MPEG規格においては、符号化ストリーム中には、フレーム構造を表すデータとして、Repeat_first_field 及びTop_field_first というフラグが設定されており、MPEGデコーダはこのフラグに基づいてデコード処理を行うので、復号化ビデオデータは30Hzのフレームレートを有したビデオデータである。

以上の説明から理解できるように、送信側システムのプロセッサにおいて、30Hzのフレームレートを有するビデオデータにアンシラリーデータを付加したとしても、MPEGエンコード処理のために必要な処理である逆3-2プルダウン処理を行うと、30Hzのフレームレートを有するビデオデータからリピートフィールドが除去されてしまう。つまり、そのリピートフィールドに付加したアンシラリーデータまでも除去されてしまうということである。従って、送出側システムから受信側システムにビデオデータを伝送する途中において、逆3-2プルダウン処理を行った場合には、その逆3-2プルダウン処理によってリピートフィールドそのものが除去されてしまうので、送出側である第1のビデオプロセッサ1から受信側である第2のビデオプロセッサ4には、そのリピートフィールドに付加したアンシラリーデータに関する情報が伝送されないという問題が発生する。

発明の開示

本発明は、入力ビデオデータを符号化するためエンコーディングシステム及び符号化ストリームをデコードするためのデコーディングシステムに関する。具体的には、MPEGエンコーディング処理及びMPEGデコーディング処理を繰り

返したとしても、ビデオデータに付加されているアンシラリーデータやビデオデータ固有の情報が無くならないように、符号化ストリームと一緒に伝送するためのシステム及び方法を提供するものである。

MPEGエンコーダは、その入力ビデオデータに付加されているのアンシラリーデータを抽出し、抽出したアンシラリーデータを符号化ストリーム中に `Ann-cillary_data()` として挿入することによって、符号化ストリームと一緒にこのアンシラリーデータを伝送する。MPEGデコーダは、符号化ストリームからアンシラリーデータを抽出し、MPEGデコーディング処理によって生成されたベースバンドビデオデータに抽出したアンシラリーデータを付加する。

入力ビデオデータをエンコードするためのエンコーディング装置において、入力ビデオデータから、ビデオデータのブランкиング期間に付加されているアンシラリーデータを抽出し、入力ビデオデータをエンコードし、符号化ストリームを生成し、符号化ストリーム中に上記アンシラリーデータが挿入されるように、上記符号化手段をコントロールする。

入力ビデオデータをエンコードすることによって生成された符号化ストリームをデコードするためのデコーディング装置において、符号化ストリームから、符号化ストリーム中に含まれていアンシラリーデータを抽出し、符号化ストリームをデコードし、復号化ビデオデータを生成し、復号化ビデオデータのブランкиング期間に、アンシラリーデータを多重化する。

入力ビデオデータを符号化することによって生成された符号化ストリームをデコードするためのデコーディング装置において、符号化ストリームのシンタックスを解析することによって、符号化ストリーム中に含まれていアンシラリーデータを得、符号化ストリームをデコードし、復号化ビデオデータを生成し、入力ビデオデータと符号化ビデオデータとが同じアンシラリーデータを有するように、復号化ビデオデータにアンシラリーデータを多重化する。

入力ビデオデータを符号化することによって生成された符号化ストリームをデ

コードするためのデコーディング装置において、符号化ストリームのピクチャエリアに含まれていアンシラリーデータを、符号化ストリームから得、符号化ストリームをデコードし、復号化ビデオデータを生成し、復号化ビデオデータとアンシラリーデータとを多重化することによって、入力ビデオデータと同じデータを生成する。

入力ビデオデータを符号化するエンコーディング手段と、符号化手段によって符号化された符号化ストリームを受け取って、符号化ストリームを復号化し復号化ビデオデータを生成するデコーディング手段とを備えたコーディングシステムにおいて、エンコーディング手段は、符号化ストリームを生成するために上記入力ビデオデータを符号化する手段と、入力ビデオデータに付与されているアンシラリーデータを符号化ストリーム中に挿入する手段とを備え、デコーディング手段は、復号化ビデオデータを生成するために符号化ストリームを復号化する手段と、符号化ストリームと共に伝送されてきたアンシラリーデータを、復号化ビデオデータに多重化する手段とを備える。

図面の簡単な説明

図1は、従来のMPEGエンコーダ及びMPEGデコーダを有するシステムの構成を示すブロック図である。

図2は、3-2プルダウン処理回路を有するエンコードシステムの構成を示すブロック図である。

図3は、3-2プルダウン処理の説明に供する略線図である。

図4は本発明によるエンコード/デコードシステムの構成を示すブロック図である。

図5は、エレメンタリストリーム及びトランSPORTストリームを示す略線図である。

図6は、MPEGエンコーダの構成を示すブロック図である。

図7は、3-2プルダウン処理の説明に供する略線図である。

図8は、ビデオデータの全画素エリア及びアクティブビデオエリアを示す略線図である。

図9は、各フレームの構造を示す略線図である。

図10は、`video sequence`のシンタックスを示す略線図である。

図11は、`sequence header`のシンタックスを示す略線図である。

図12は、`sequence extension`のシンタックスを示す略線図である。

図13は、`extension and user data`のシンタックスを示す略線図である。

図14は、`user data`のシンタックスを示す略線図である。

図15は、`data ID`のシンタックスを示す略線図である。

図16は、`V-Phase`のシンタックスを示す略線図である。

図17は、`H-Phase`のシンタックスを示す略線図である。

図18は、`time code`のシンタックスを示す略線図である。

図19は、`time code`のシンタックスを示す略線図である。

図20は、`picture order`のシンタックスを示す略線図である。

図21は、`ancillary data`のシンタックスを示す略線図である。

図22は、`group of picture header`のシンタックスを示す略線図である。

図23は、`picture header`のシンタックスを示す略線図である。

図24は、`picture coding extension`のシンタックスを示す略線図である。

図25は、`picture data`のシンタックスを示す略線図である。

図26は、シーケンスレイヤ、GOPレイヤ、ピクチャレイヤの各データを示す略線図である。

図27は、エンコーダ側の多重化部の構成を示すブロック図である。

図28は、ソースビデオデータからPESパケット及びTSパケットを生成する方法の説明に供する略線図である。

図29は、PESヘッダのデータ構成を示す略線図である。

図30は、各ピクチャの配列を示す略線図である。

図31は、MPEGデコーダの構成を示すブロック図である。

発明を実施するための最良の形態

図4は、メイン放送局141及び地方放送局171からなる放送システムを表している図である。

メイン放送局141には、複数のエディティング/プロセッシングスタジオ145A～145D、複数のMPEGエンコーダ142A～142Dと、複数のMPEGデコーダ144A～144Dと、少なくとも1つのマルチプレクサ162Aと少なくとも1つのデマルチプレクサ161Aとを備えている。また、この放送局内141には、SMPTE305Mとして定義されているSDTI-CP (Serial Data Transfer Interface-Content Package) ネットワーク150を備え、各MPEGエンコーダ、各MPEGデコーダ、マルチプレクサ162A及びデマルチプレクサ161Aが、SDTI-CPネットワーク150を介してエレメンタリーストリームを送受信することができる。SDTI-CPは、MPEGのエレメンタリーストリームを伝送するために提案された通信フォーマットであって、SMPTE305Mとして定義されている。尚、このSDTI-CPネットワーク150上に転送されたエレメンタリーストリームは、ES_over_SD TI-CPと表現されている。

エディティング/プロセッシングスタジオ145A～145Dは、ビデオサー

バー、ビデオ編集機、特殊効果装置及びビデオスイッチャなどを備えているスタジオである。MPEGデコーダから復号化されたベースバンドのビデオデータを受け取り、そのベースバンドのビデオデータに対して、編集や画像処理を行ったあと、編集ビデオデータ又は画像処理ビデオデータを、MPEGエンコーダに出力する。すなわち、エディティング/プロセッシングスタジオ145A～145Dは、符号化ストリームのストリームプロセッシングではなく、ベースバンドビデオ信号の編集や画像処理を行うためのプロセッシングユニットである。

MPEGエンコーダ142A～142Dは、エディティング/プロセッシングスタジオ145A～145Dからベースバンドの入力ビデオ信号を受け取り、その入力ビデオデータを、前述したMPEG規格に従ってエンコードし、エレメンタリーストリーム(ES)を生成するための装置である。MPEGエンコーダによって生成されたエレメンタリーストリームは、SDTI-CPネットワーク150を介して、MPEGデコーダ144A～144DのうちのいずれかのMPEGデコーダ又はマルチプレクサ162Aに供給される。

MPEGデコーダ144A～144Dは、MPEGエンコーダ142A～142D又はデマルチプレクサ161Aから、SDTI-CPネットワーク150を介して供給されたエレメンタリーストリームを受け取り、MPEG規格に基づいてデコード処理を行う。マルチプレクサ162Aは、メイン放送局内141において制作された複数のビデオプログラムを地方局171又は各家庭に配信するために、多重化することによって一つのトランSPORTストリームを生成するための回路である。具体的には、SDTI-CPネットワーク150を介して複数のビデオプログラムに対応する複数のエレメンタリーストリーム受けとり、各エレメンタリーストリームをパケット化することによってパケット化ドエレメンタリーストリームを生成し、そのパケット化ドエレメンタリーストリームからトランSPORTストリームパケットを生成する。複数のビデオプログラムから生成されたトランSPORTストリームパケットを多重化することによって、マルチプレクサ162Aは多重化されたトランSPORTストリームを生成する。尚、こ

のマルチプレクサ 162A の構成及び処理については、後述する。

複数のビデオプログラムの伝送先が地方局 171 の場合には、マルチプレクサ 162A によって生成された多重化トранSPORTストリームは、地方局 171 のマルチプレクサ 161B に ATM や衛星回線などのネットワークを介して供給される。地方局 171 は、規模こそメイン局 141 と異なるが、全く同じシステム構成を有しているので、詳細については説明を省略する。

複数のビデオプログラムの伝送先が地方局 171 の場合には、マルチプレクサ 162A によって生成された多重化トランSPORTストリームは、各家庭のセットトップボックス内に設けられている MPEG デコーダ 170A に、ATM や衛星回線などのネットワークを介して供給され、デコードされたビデオデータが TV セットに供給される。

図 5 は、放送局内において SDTI-CP ネットワークを介して伝送されるエレメンタリーストリームと、公衆ネットワークを介して伝送されるトランSPORTストリームとの違いを表している。

放送局内においては、エレメンタリーストリームは、SDTI-CP ネットワークを使用して伝送される。この SDTI-CP ネットワーク 150 は、SMPTE 259M によって規格化されている SDI (Serial Data Interface) をベースとした 270Mbps の伝送速度を可能にしている通信フォーマットであって、MPEG 方式のエレメンタリーストリーム (ES) をそのまま伝送することが可能であり、スタジオ内のような閉じたネットワークに適している。具体的には、図 5A に示すように、ビデオデータのフレーム単位で、ビデオデータ「V」とオーディオデータ「A」がパッキングされており、フレームシンク (点線) により区切られたフレーム境界で、簡単に編集を行うことができる。

放送局間の伝送や公衆ネットワークを使用する際には、トランSPORTストリームの形態でビデオデータが伝送される。トランSPORTストリームは、伝送容量の少ない公衆ネットワークにおいてもデータ転送を可能にするために、図 5B

に示すように、ビデオデータやオーディオなどのすべてのコンテンツは、188 バイトの単位にパッキングされて伝送される。同様に、「V」はビデオデータのトランSPORTストリームパケットと示し、「A」はオーディオデータのトランSPORTストリームパケットを示し、空白は空きデータのパケットを示している。

次に、図6を参照して、MPEGエンコーダ142A～142Dについて説明する。まず最初に、このMPEGエンコーダに供給される入力ビデオデータについて補足する。この実施例においては、この入力ビデオデータは、24Hzのフレームレートを有していたソースビデオデータに対して、3-2プルダウン処理を行うことによって生成された30Hzのビデオデータである。具体的には、24Hzのフレームレートを有したオリジナルのソースビデオデータは、各フレームF1、F2、……がそれぞれ2つのフィールド（トップフィールドt1、t2、……及びボトムフィールドb1、b2、……）から構成されいるビデオデータであって、3-2プルダウン処理では、図7Aに示すように、トップフィールドファーストである第1のフレームF1ではトップフィールドt1を繰り返してリピートフィールドt1'を生成することにより3フィールドで1つのフレームを構成し、また、ボトムフィールドファーストである第3のフレームF3ではボトムフィールドb3を繰り返してリピートフィールドb3'を生成する。このようにして、3-2プルダウン処理では、3フィールド構成のフレームと2フィールド構成のフレームを交互に繰り返すことにより、フレームレートが24Hzのソースビデオデータから、図7Aのようなフレームレートが30Hzのビデオデータを生成することができる。

尚、この実施例では、入力ビデオデータが、3-2プルダウン処理によって生成された入力ビデオデータの例を示しているが、本発明は、入力ビデオデータが3-2プルダウンされたビデオデータの実施例に限られるものではなく、オリジナルソースビデオが30Hzのフレームレートを有するビデオデータであって、3-2プルダウン処理が施されていないビデオデータの場合であっても問題無く適用できるものである。

図6に示されたMPEGエンコーダは、補助データ分離回路101、フィールドカウンタ102、逆3-2プルダウン回路103、エンコーディングコントローラー104、動きベクトル検出回路105、スイッチ回路111、DCT回路112、量子化回路113、逆量子化回路114、逆DCT回路115、加算回路116、メモリ117及び118、動き補償回路119、演算回路120、121、122、可変長符号化回路125及び送信バッファ126を備えている。

補助データ分離回路101は、入力ビデオデータのプランキング期間から、アンシラリーデータを抽出する回路である。具体的には、図8に示したように、入力ビデオデータの垂直プランキング期間に挿入されているアンシラリーデータと、そのアンシラリーデータが挿入されているライン番号を抽出する。このアンシラリーデータとしては、テレテキストデータ、クローズドキャプションデータ、SMPTE RP164において定義されているVITC (Vertical Interval Time Code) やRP196において定義されているLTC (Linear Time Code) 等であるが、これのみに限定する必要はない。本実施例においては、抽出したアンシラリーデータに関する情報は、Ancillary_dataとしてコントローラ104に供給されるとともに、ライン番号に関する情報は、Line_numberとしてコントローラ104に供給される。また、VITCに関する情報は、Time_code_1としてコントローラ104に供給され、LTCに関する情報は、Time_code_2としてコントローラ104に供給される。

また、補助データ分離回路101は、入力ビデオデータが有する固有情報を入力ビデオデータから抽出する。この固有情報とは、例えば、図8に示したように、入力ビデオデータの全画素エリアAR1に対するアクティブビデオエリアAR2が存在する位置を示すデータであり、具体的には、アクティブビデオエリアの垂直方向の開始位置を示すライン数及び水平方向の開始位置を示すサンプル数などである。本実施例においては、アクティブビデオエリアの垂直方向の位置に関する情報は、V-phase アクティブビデオエリアの水平方向の位置に関する情報は、V-phase

る情報は、H-phase としてコントローラ 104 に供給される。その他の固有情報の例としては、入力ビデオデータに付与されたソース名称、撮影場所、撮影時間などである。

補助データ分離部 101 から出力された入力ビデオデータは、続くフィールドカウンタ 102 に供給される。フィールドカウンタ 102 は、入力ビデオデータの各フレームを構成するフィールドを各フレームごとにカウントする回路である。フィールドカウンタ 102 は、そのフレーム毎のカウント情報を、Field_ID としてコントローラ 104 に供給する。たとえば、図 7A に示したような入力ビデオデータがこのフィールドカウンタ 102 に供給された場合には、フレーム F1 には 3 つのフィールドが存在するので、カウント情報として、「0」、「1」及び「2」の Field_ID が出力され、フレーム F2 には 2 つのフィールドが存在するので、カウント情報として、「0」及び「1」の Field_ID が出力される。

さらに、このフィールドカウンタ 102 は、入力ビデオデータのフィールドをカウントすることによってカウントアップされる 2 つのカウンタを備え、このカウンタによるカウント値に関する情報を PTS_counter 及び DTS_counter としてコントローラ 104 に出力する。PES ヘッダを生成するときに、PTS_counter はプレゼンテーションタイムスタンプ (PTS) を生成するための情報として使用され、DTS_counter はデコーディングタイムスタンプ (DTS) を生成するための情報として使用される。

この PTS_counter 及び DTS_counter について、図 9 を参照して詳細に説明する。図 9 は、入力ビデオデータの各フレームのフレーム構造と、各フレームにおける PTS_counter 及び DTS_counter の関係を示している図である。図 9 について詳細に説明する前に、Repeat_first_field 及び Top_field_first フラグについて補足しておく。Repeat_first_field のフラグの “1” は、MPEG デコード時に、リピートフィールドを作成する必要があることを意味し、

`Repeat_first_field`のフラグの”0”は、MPEGデコード時に、リピートフィールドを作成する必要がないことを意味する。`Top_field_first`のフラグは、フレームを構成するフィールドのうち、最初のフィールドがトップフィールドであるのか、またはボトムフィールドであるのかを表している。`Top_field_first`フラグの”1”は、トップフィールドがボトムフィールドより時間的に早いフレーム構造であることを表しており、`Top_field_first`フラグの”0”は、ボトムフィールドがトップフィールドより時間的に早いフレーム構造であることを表している。図9Aは、図7Aにおいて説明した入力ビデオデータに関するフレーム構造について説明するための図である。具体的には、最初のフレームF1をデコードする際に、単純にトップフィールドとボトムフィールドからなるフレームを生成するのではなく、トップフィールドをコピーしてリピートフィールドを作成することで、3フィールドからなるフレームを生成する必要がある。従って、対応する`Repeat_first_field`のフラグは”1”となり、`Top_field_first`のフラグは”1”となる。

フレームF2のデコード処理においては、リピートフィールドを生成する必要がないので、`Repeat_first_field`のフラグは”0”とされ、ボトムフィールドがトップフィールドより時間的に早いフレームであるため、`Top_field_first`のフラグは”0”とされる。

フレームF3のデコード処理においては、そのボトムフィールドをコピーしてリピートフィールドが作成され、符号化フレームが3フィールドに変換されなければいけない。従って、`Repeat_first_field`のフラグは”1”とされ、`Top_field_first`のフラグは”0”とされる。フレームF4のデコード処理において、リピートフィールドが作成する必要はないので、`Repeat_first_field`のフラグは”0”とされ、`Top_field_first`のフラグは1とされる。さて、上述したように、`PTS_counter`はPTSの基になるタイムスタンプ情報であるので、入力ビ

デオデータのフレーム順と一致している情報でなければいけない。具体的には、このPTS_counterは、0から127まで増加した後、再び0に戻るカウント動作を行うカウンターによって生成される値である。従って、カウンタPTS_counterの値は、図9Bに示すように変化する。具体的に説明すると、入力ビデオデータにおいて、最初のフレームF1はIピクチャであり、最初に表示しなければいけないピクチャであるので、PTS_counterの値は”0”である。2番目のフレームF2のPTS_counterの値は、フレームF1のPTS_counterの値”0”に、フレームF1に含まれているフィールド数”3”を加算した値”3”($=0+3$)となる。3番目のフレームF3のPTS_counterの値は、フレームF2のPTS_counterの値”3”に、フレームF2に含まれているフィールド数”2”を加算した値”5”($=3+2$)となる。4番目のフレームF4のPTS_counterの値は、フレームF3のPTS_counterの値”5”に、フレームF2に含まれているフィールド数”3”を加算した値”8”($=5+3$)となる。フレームF5以降のPTS_counterの値も同様に算出される。さらに、DTS_counterは、DTSの基になるタイムスタンプ情報であるので、入力ビデオデータのフレーム順では無くて、符号化ストリーム中のピクチャ順と一致する情報でなければいけない。図9Cを参照して、具体的に説明すると、最初のフレームF1は、Iピクチャであるので、フレームF1が表示される表示タイミングよりフレームF1がデコードされるタイミングが1フレーム分だけ早くなくてはいけない。すなわち、フレームF1の前のフレームF0は2つのフィールドから構成されているため、DTS_counterの値は、表示タイムスタンプPTS_counter=0を基準時間としたとき、基準時間”0”より”3”フィールド分だけ前の値”125”となる。尚、DTS_counterは27($=128$)のモジュロで表されるため、その値は0から127の間の値を循環する。フレームF1の次に符号化されるフレームF4のDTS_counterの値は、フレームF1のDTS_counterの値”125”に、フレームF1のフ

フィールド数“3”を加えた値“0”($=128=125+3$)となる。次に符号化されるフレームF2はBピクチャであるため、DTS_counterの値は、PTS_counterと同じ値になり、その値は“3”となる。同様に、次に符号化されるフレームF3もBピクチャであるので、DTS_counterの値は、PTS_counterの値と同一とされ、その値は“5”とされる。以下、フレームF7以降のDTS_counterの値も、同様に算出されるので、ここでは、その説明を省略する。フィールドカウンター102は、上述した決まりに従って、PTS_counter及びDTS_counterを生成し、コントローラ104に供給する。

逆3-2プルダウン回路103は、フィールドカウンタ102から出力されたビデオデータを受け取り、逆3-2プルダウン処理を行う。この逆3-2プルダウン処理回路103は、図7Aに示すような3-2プルダウン処理により得られた30Hzのフレームレートのビデオデータを受け取って、24Hzのフレームレートのビデオデータを生成するための回路である。具体的には、逆3-2プルダウン処理回路7は、図7Bに示すように、3-2プルダウン処理によって挿入されたリピートフィールドt1'及びb3'を除去することによって、30Hzのフレームレートのビデオデータを、24Hzのフレームレートのビデオデータに変換する。逆プルダウン処理回路103は、リピートフィールドを除去する際に、供給されたビデオデータのフレーム構造を解析し、所定間隔で出現するリピートフィールドであると認定したフィールドのみを除去するようにしている。よって、逆プルダウン処理回路103は、ビデオデータのフレーム構造を解析したときに、フレーム構造を表す情報として、Repeat_first_fieldとTop_field_firstのフラグを生成し、コントローラ104に供給する。

動きベクトル検出回路105は、逆3-2プルダウン処理回路103から出力されたビデオデータを、マクロブロック単位で受け取り、予め設定されている所定のシーケンスに従って、各フレームの画像データを、Iピクチャ、Pピクチ

ヤ、又はBピクチャとして処理する。シーケンシャルに入力される各フレームの画像を、I、P又はBのいずれのピクチャとして処理するかは、オペレータによって指定されるGOP構造に基づいて予め定められている。検出された動きベクトルMVは、コントローラ104及び動き補償回路119に供給される。スイッチ回路111は、画像内予測モードが設定されたとき、スイッチを接点aに切り換える。従って、このマクロブロックデータは、Iピクチャのデータと同様に、DCT回路112、量子化回路113、可変長符号化回路125及び送信バッファ126を介して伝送路に伝送される。また、この量子化データは、逆量子化回路114、逆DCT回路115、及び演算器116を介しての後方予測画像用のフレームメモリ117に供給される。また、前方予測モードが設定された場合、スイッチ111が接点bに切り換えられると共に、の前方予測画像用のフレームメモリ118に記憶されている画像データ（この場合Iピクチャの画像データ）が読み出され、動き補償回路119により、動きベクトル検出回路105がから供給されている動きベクトルMVに基づいて動き補償が行われる。すなわち、動き補償回路119は、前方予測モードの設定が指令されたとき、前方予測画像用のメモリ118の読み出しアドレスを、動きベクトル検出回路105が、現在出力しているマクロブロックの位置に対応する位置から動きベクトルに対応する分だけずらしてデータを読み出し、予測画像データを生成する。動き補償回路119より出力された予測画像データは、演算器120に供給される。演算器120は、参照画像のマクロブロックのデータから、動き補償回路119より供給された、このマクロブロックに対応する予測画像データを減算し、その差分（予測誤差）を出力する。この差分データは、DCT回路112、量子化回路113、可変長符号化回路125及び送信バッファ126を介して伝送路に伝送される。また、この差分データは、逆量子化回路114、及びIDCT回路115により局所的に復号され、演算器116に入力される。この演算器116にはまた、演算器120に供給されている予測画像データと同一のデータが供給されている。演算器116は、IDCT回路115が出力する差分データに、動き補償回路119

が output する予測画像データを加算する。これにより、元の（復号した）Pピクチャの画像データが得られる。このPピクチャの画像データは、フレームメモリの後方予測画像部117に供給されて記憶される。動きベクトル検出回路105は、このように、IピクチャとPピクチャのデータが前方予測画像部118と後方予測画像部117にそれぞれ記憶された後、次にBピクチャの処理を実行する。画像内予測モード又は前方予測モードの時、スイッチ111は接点a又はbに切り換えられる。このとき、Pピクチャにおける場合と同様の処理が行われ、データが伝送される。これに対して、後方予測モード又は両方向予測モードが設定されたとき、スイッチ111は接点c又はdにそれぞれ切り換えられる。スイッチ111が接点cに切り換えられている後方予測モードのとき、後方予測画像部117に記憶されている画像（現在の場合、Pピクチャの画像）データが読み出され、動き補償回路119により、動きベクトル検出回路105が output する動きベクトルに対応して動き補償される。すなわち、動き補償回路119は、後方予測モードの設定が指定されたとき、後方予測画像部117の読み出しアドレスを、動きベクトル105が、現在出力しているマクロブロックの位置に対応する位置から動きベクトルに対応する分だけずらしてデータを読み出し、予測画像データを生成する。動き補償回路119より出力された予測画像データは、演算器121に供給される。演算器121は、参照画像のマクロブロックのデータから、動き補償回路119より供給された予測画像データを減算し、その差分を出力する。この差分データは、DCT回路112、量子化回路113、可変長符号化回路125及び送信バッファ126を介して伝送路に伝送される。スイッチ111が接点dに切り換えられている両方向予測モードのとき、前方予測画像部118に記憶されている画像（現在の場合、Iピクチャの画像）データと、後方予測画像部117に記憶されている画像（現在の場合、Pピクチャの画像）データが読み出され、動き補償回路119により、動きベクトル検出回路105が output する動きベクトルに対応して動き補償される。すなわち、動き補償回路119は、両方向予測モードの設定が指定されたとき、前方予測画像部118と後方予測画像部

117の読み出しアドレスを、動きベクトル検出回路105がいま出力しているマクロブロックの位置に対応する位置から動きベクトル（この場合の動きベクトルは、前方予測画像用と後方予測画像用の2つとなる）に対応する分だけずらしてデータを読み出し、予測画像データを生成する。動き補償回路119より出力された予測画像データは、演算器122に供給される。演算器122は、動きベクトル検出回路105より供給された参照画像のマクロブロックのデータから、動き補償回路119より供給された予測画像データの平均値を減算し、その差分を出力する。この差分データは、DCT回路112、量子化回路113、可変長符号化回路125及び送信バッファ126を介して伝送路に伝送される。Bピクチャの画像は、他の画像の予測画像とされることがないため、フレームメモリ117、118には記憶されない。コントローラ104は、上述した予測モードに関する処理、DCTモードに関する処理及び量子化に関する処理に携わる全ての回路をコントロールする。さらには、コントローラ104は、各ピクチャのエンコーディング処理において発生した、動きベクトル、ピクチャタイプ、予測モード、DCTモード、量子化ステップなどの全てのコーディングパラメータを、可変長符号化回路125に供給する。

また、コントローラ104は、補助データ分離回路101から、V-phase、H-phase、Time_code1、Time_code2、Ancillary_data及びLine_numberに関する情報を受け取ると共に、フィールドカウンタ102からDTS_counter PTS_counter及びField_IDに関する情報を受け取る。コントローラ104は、受け取った、V-phase、H-phase、Time_code1、Time_code2、Ancillary_data、Line_number、DTS_counter PTS_counter及びField_IDに関する情報を、MPEG_ES_editing_information (i) として可変長符号化回路125に供給する。

可変長符号化回路125は、量子化回路113から量子化されたDCT係数

及びコントローラ 104 から供給されたコーディングパラメータを可変長符号化すると共に、MPEG 規格において定められているエレメンタリーストリームのシンタックスに従って、符号化ストリームを生成する。

実施例の特徴ある点として、さらに、可変長符号化回路 125 は、コントローラ 104 から `MPEG_ES_editing_information (i)` として供給された情報を可変長符号化し、符号化ストリーム中に挿入する処理を行う。符号化ストリームのシンタックス及び `MPEG_ES_editing_information (i)` のシンタックスについては、詳しくは後述する。

次に、図 10 乃至図 26 を参照してビットストリームのシンタックスについて説明する。尚、図 26 は、図 10 から図 25 において説明している詳細なシンタックスを、より理解しやすい形式で MPEG 符号化ストリームのデータ構造を示している説明図である。図 10 は、MPEG のビデオストリームのシンタックスを表わした図である。MPEG エンコーダ 42 は、この図 10 に示されたシンタックスに従った符号化エレメンタリーストリームを生成する。以下に説明するシンタックスにおいて、関数や条件文は細活字で表わされ、データエレメントは、太活字で表されている。データ項目は、その名称、ビット長およびそのタイプ・伝送順序を示すニーモニック (Mnemonic) で記述されている。まず、この図 10 に示されているシンタックスにおいて使用されている関数について説明する。実際には、この図 10 に示されているシンタックスは、MPEG デコーダ 44 側において、伝送されてきた符号化ビットストリームから所定の意味のあるデータエレメントを抽出するために使用されるシンタックスである。MPEG エンコーダ 42 側において使用されるシンタックスは、図 10 に示されたシンタックスから `if` 文や `while` 文等の条件文を省略したシンタックスである。

`video_sequence ()` において最初に記述されている `next_start_code ()` は、ビットストリーム中に記述されているスタートコードを探すための関数である。この図 10 に示されたシンタックスに従って生成さ

れた符号化ストリームには、まず最初に、`sequence_header()` と `sequence_extension()` によって定義されたデータエレメントが記述されている。この `sequence_header()` は、MPEG ビットストリームのシーケンスレイヤのヘッダデータを定義するための関数であつて、`sequence_extension()` は、MPEG ビットストリームのシーケンスレイヤの拡張データを定義するための関数である。`sequence_extension()` の次に配置されている `do { } while` 構文は、`while` 文によって定義されている条件が真である間、`do` 文の { } 内の関数に基いて記述されたデータエレメントが符号化データストリーム中に記述されていることを示す構文である。この `while` 文に使用されている `next_bits()` は、ビットストリーム中に記述されているビット又はビット列と、参照されるデータエレメントとを比較するための関数である。この図 10 に示されたシンタックスの例では、`next_bits()` は、ビットストリーム中のビット列とビデオシーケンスの終わりを示す `sequence_end_code` とを比較し、ビットストリーム中のビット列と `sequence_end_code` とが一致しないときに、この `while` 文の条件が真となる。従つて、`sequence_extension()` の次に配置されている `do { } while` 構文は、ビットストリーム中に、ビデオシーケンスの終わりを示す `sequence_end_code` が現れない間、`do` 文中の関数によって定義されたデータエレメントが符号化ビットストリーム中に記述されていることを示している。符号化ビットストリームにおいて、`sequence_extension()` によって定義された各データエレメントの次には、`extension_and_user_data(0)` によって定義されたデータエレメントが記述されている。この `extension_and_user_data(0)` は、MPEG ビットストリームのシーケンスレイヤにおける拡張データとユーザデータを定義するための関数である。この `extension_and_user_data(0)` の次に配置されている `do { } while` 構文は、`w`

h i l e 文によって定義されている条件が真である間、 d o 文の { } 内の関数に基いて記述されたデータエレメントが、ビットストリームに記述されていることを示す関数である。このw h i l e 文において使用されているn e x t b i t s () は、ビットストリーム中に現れるビット又はビット列と、 p i c t u r e _ s t a r t _ c o d e 又はg r o u p _ s t a r t _ c o d e との一致を判断するための関数であるって、ビットストリーム中に現れるビット又はビット列と、 p i c t u r e _ s t a r t _ c o d e 又はg r o u p _ s t a r t _ c o d e とが一致する場合には、 w h i l e 文によって定義された条件が真となる。よって、このd o { } w h i l e 構文は、符号化ビットストリーム中において、 p i c t u r e _ s t a r t _ c o d e 又はg r o u p _ s t a r t _ c o d e が現れた場合には、そのスタートコードの次に、 d o 文中の関数によって定義されたデータエレメントのコードが記述されていることを示している。このd o 文の最初に記述されている i f 文は、符号化ビットストリーム中にg r o u p _ s t a r t _ c o d e が現れた場合、という条件を示しいる。この i f 文による条件は真である場合には、符号化ビットストリーム中には、このg r o u p _ s t a r t _ c o d e の次にg r o u p _ o f _ p i c t u r e _ h e a d e r () およびe x t e n s i o n _ a n d _ u s e r _ d a t a (1) によって定義されているデータエレメントが順に記述されている。このg r o u p _ o f _ p i c t u r e _ h e a d e r () は、M P E G 符号化ビットストリームのG O P レイヤのヘッダデータを定義するための関数であって、 e x t e n s i o n _ a n d _ u s e r _ d a t a (1) は、M P E G 符号化ビットストリームのG O P レイヤの拡張データおよびユーザデータを定義するための関数である。さらに、この符号化ビットストリームにおいて、 g r o u p _ o f _ p i c t u r e _ h e a d e r () およびe x t e n s i o n _ a n d _ u s e r _ d a t a (1) によって定義されているデータエレメントの次には、 p i c t u r e _ h e a d e r () とp i c t u r e _ c o d i n g _ e x t e n s i o n () によって定義されたデータエレメントが記述されている。もちろん、先に説明した i f 文

の条件が真とならない場合には、group_of_picture_header () およびextension_and_user_data (1) によって定義されているデータエレメントは記述されていないので、extension_and_user_data (0) によって定義されているデータエレメントの次に、picture_header ()、picture_coding_extension () およびextension_and_user_data (2) によって定義されたデータエレメントが記述されている。このpicture_header () は、MPEG符号化ビットストリームのピクチャレイヤのヘッダデータを定義するための関数であって、picture_coding_extension () は、MPEG符号化ビットストリームのピクチャレイヤの第1の拡張データを定義するための関数である。extension_and_user_data (2) は、MPEG符号化ビットストリームのピクチャレイヤの拡張データおよびユーザデータを定義するための関数である。このextension_and_user_data (2) によって定義されるユーザデータは、ピクチャレイヤに記述されているデータであって、各ピクチャ毎に記述することのできるデータである。符号化ビットストリームにおいて、ピクチャレイヤのユーザデータの次には、picture_data () によって定義されるデータエレメントが記述されている。このpicture_data () は、スライスレイヤおよびマクロブロックレイヤに関するデータエレメントを記述するための関数である。このpicture_data () の次に記述されているwhile文は、このwhile文によって定義されている条件が真である間、次のif文の条件判断を行うための関数である。このwhile文において使用されているnextbits () は、符号化ビットストリーム中に、picture_start_code又はgroup_start_codeが記述されているか否かを判断するための関数であって、ビットストリーム中にpicture_start_code又はgroup_start_codeが記述されている場合には、このwhile 文によって定義された条件が真となる。

次の if 文は、符号化ビットストリーム中に sequence_end_code が記述されているか否かを判断するための条件文であって、 sequence_end_code が記述されていないのであれば、 sequence_header() と sequence_extension() とによって定義されたデータエレメントが記述されていることを示している。 sequence_end_code は符号化ビデオストリームのシーケンスの終わりを示すコードであるので、符号化ストリームが終了しない限り、符号化ストリーム中には sequence_header() と sequence_extension() とによって定義されたデータエレメントが記述されている。この sequence_header() と sequence_extension() によって記述されたデータエレメントは、ビデオストリームのシーケンスの先頭に記述された sequence_header() と sequence_extension() によって記述されたデータエレメントと全く同じである。このように同じデータをストリーム中に記述する理由は、ビットストリーム受信装置側でデータストリームの途中（例えばピクチャレイヤに対応するビットストリーム部分）から受信が開始された場合に、シーケンスレイヤのデータを受信できなくなり、ストリームをデコード出来なくなることを防止するためである。この最後の sequence_header() と sequence_extension() とによって定義されたデータエレメントの次、つまり、データストリームの最後には、シーケンスの終わりを示す2ビットの sequence_end_code が記述されている。以下に、 sequence_header()、 sequence_extension()、 extension_and_user_data(0)、 group_of_picture_header()、 picture_header()、 picture_coding_extension()、および picture_data() について詳細に説明する。図11は、 sequence_header() のシンタックスを説明するための図である。この sequence_header() によって定義されたデータエレメントは、 s

`sequence_header_code`、`horizontal_size_value`、`vertical_size_value`、`aspect_ratio_information`、`frame_rate_code`、`bit_rate_value`、`marker_bit`、`vbv_buffer_size_value`、`constrained_parameter_flag`、`load_intra_quantizer_matrix`、`intra_quantizer_matrix [64]`、`load_non_intra_quantizer_matrix`、および`non_intra_quantizer_matrix`等である。`sequence_header_code`は、シーケンスレイヤのスタート同期コードを表すデータである。`horizontal_size_value`は、画像の水平方向の画素数の下位12ビットから成るデータである。`vertical_size_value`は、画像の縦のライン数の下位12ビットからなるデータである。`aspect_ratio_information`は、画素のアスペクト比（縦横比）または表示画面アスペクト比を表すデータである。`frame_rate_code`は、画像の表示周期を表すデータである。`bit_rate_value`は、発生ビット量に対する制限のためのビット・レートの下位18ビット（400 bsp 単位で切り上げる）データである。`marker_bit`は、スタートコードエミュレーションを防止するために挿入されるビットデータである。`vbv_buffer_size_value`は、発生符号量制御用の仮想バッファ（ビデオバッファアベリファイヤー）の大きさを決める値の下位10ビットデータである。`constrained_parameter_flag`は、各パラメータが制限以内であることを示すデータである。`load_intra_quantizer_matrix`は、イントラMB用量子化マトリックス・データの存在を示すデータである。`intra_quantizer_matrix [64]`は、イントラMB用量子化マトリックスの値を示すデータである。`load_non_intra_quantizer_matrix`は、非イントラMB用量子化マトリックス・データ

タの存在を示すデータである。`non_intra_quantizer_matrix`は、非イントラMB用量子化マトリックスの値を表すデータである。図12は`sequence_extension()`のシンタックスを説明するための図である。この`sequence_extension()`によって定義されたデータエレメントとは、`extension_start_code`、`extension_start_code_identifier`、`profile_and_level_indication`、`progressive_sequence`、`chroma_format`、`horizontal_size_extension`、`vertical_size_extension`、`bit_rate_extension`、`vbv_buffer_size_extension`、`low_delay`、`frame_rate_extension_n`、および`frame_rate_extension_d`等のデータエレメントである。

`extension_start_code`は、エクステンションデータのスタート同期コードを表すデータである。`extension_start_code_identifier`は、どの拡張データが送られるかを示すデータである。`profile_and_level_indication`は、ビデオデータのプロファイルとレベルを指定するためのデータである。`progressive_sequence`は、ビデオデータが順次走査であることを示すデータである。`chroma_format`は、ビデオデータの色差フォーマットを指定するためのデータである。`horizontal_size_extension`は、シーケンスヘッダの`horizontal_size_value`に加える上位2ビットのデータである。`vertical_size_extension`は、シーケンスヘッダの`vertical_size_value`加える上位2ビットのデータである。`bit_rate_extension`は、シーケンスヘッダの`bit_rate_value`に加える上位12ビットのデータである。`vbv_buffer_size_extension`は、シーケンスヘッ

ダの `vbv_buffer_size_value` に加える上位 8 ビットのデータである。`low_delay` は、B ピクチャを含まないことを示すデータである。`frame_rate_extension_n` は、シーケンスヘッダの `frame_rate_code` と組み合わせてフレームレートを得るためのデータである。`frame_rate_extension_d` は、シーケンスヘッダの `frame_rate_code` と組み合わせてフレームレートを得るためのデータである。図 13 は、`extension_and_user_data(i)` のシンタックスを説明するための図である。この `extension_and_user_data(i)` は、「i」が 1 以外のときは、`extension_data()` によって定義されるデータエレメントは記述せずに、`user_data()` によって定義されるデータエレメントのみを記述する。よって、`extension_and_user_data(0)` は、`user_data()` によって定義されるデータエレメントのみを記述する。まず、図 13 に示されているシンタックスにおいて使用されている関数について説明する。`nextbits()` は、ビットストリーム中に現れるビットまたはビット列と、次に復号されるデータエレメントとを比較するための関数である。図 14 の `user_data()` は、本実施例の特徴のあつ点を説明している図である。図 14 に示すように、`user_data()` は、`user_data_start_code`, `V-phase()`, `H-phase()`, `Time_code()`, `Picture_order()`, `Ancillary_data()`, `history_data()`, および `user_data()` に関するデータエレメントを記述するための関数である。`user_data_start_code` は、MPEG 方式のビットストリームのピクチャレイヤのユーザデータエリアの開始を示すためのスタートコードである。この `user_data_start_code` の次に記述されている `if` 文は、`user_data(i)` の `i` が “0” のとき、次に記述されている `while` 構文を実行する。この `while` 構文は、ビットストリーム中に、23 個の “0” とそれに続く “1” から構成される

24ビットのデータが現れない限り真となる。この23個の”0”とそれに続く”1”から構成される24ビットのデータは、すべてのスタートコードの先頭に付与されるデータであって、すべてのスタートコードは、この24ビットの後ろに設けられることによって、`nextbits()`は、ビットストリーム中において、各スタートコードの位置を見つけることができる。`while`構文が真のとき、`user_data(i)`の*i*が”0”のときをしめしているので、これはシーケンスレイヤの`extnition_and_user_data(0)`を示すことになる。つまり、図26において、シーケンスレイヤにおいて、`extnition_and_user_data(0)`205に関するデータエレメントが記述されていることを示している。その次に記述されている`if`文の`nextbits()`は、`V-Phase`を示すビット列(`Data_ID`)を検出すると、そのビット列(`Data_ID`)の次ビットから`V-Phase()`で示される`V-Phase`のデータエレメントが記述されていることを知る。次の`Else if`文の`nextbits()`は、`H-Phase`を示すビット列(`Data_ID`)を検出すると、そのビット列(`Data_ID`)の次ビットから`H-Phase()`で示される`H-Phase`のデータエレメントが記述されていることを知る。

つまり、図26に示したように`V-Phase()`220及び`H-Phase()`221に関するデータエレメントは、シーケンスレイヤのユーザデータエリアに記述されているということを意味する。ここで、図15に示すように、`V-Phase`の`Data_ID`は、”01”を表すビット列であり、`H-Phase`の`Data_ID`は、”02”を表すビット列である。ビットストリームに記述される`V-Phase()`のシンタックスについて、図16を参照して説明する。まず、`Data_ID`は、前述したように、その`Data_ID`の次のビット列のデータエレメントが`V-Phase`であることを表す8ビットのデータであり、図15で示した値”01”である。`V-Phase`は、画像信号のフレームにおいて、符号化される最初のラインを示す16ビットのデータである。言いか

えると、V-Phaseは、アクティブビデオエリアの垂直方向のライン位置を示すデータである。

ビットストリームに記述されるH-Phase()のシンタックスについて、図17を参照して説明する。まず、Data_IDは、前述したように、そのData_IDの次のビット列のデータエレメントがH-Phaseであることを表す8ビットのデータであり、図15で示した値”02”である。H-Phaseは、画像信号フレームにおいて、符号化される最初のサンプルを示す8ビットのデータである。言いかえると、H-Phaseは、アクティブビデオエリアの水平方向の画素サンプル位置を示すデータである。

図14に戻って、次のElse if文は、extntion_and_user_data(i)のiが2のとき、次に記述されているwhile構文を実行する。while構文の内容は前述した場合と同様であるので、ここではその説明を省略する。while構文が真のとき、次のif文において、next_bits()は、Time_code1を示すビット列を検出するか、または、Time_code2を示すビット列を検出すると、そのビット列の次ビットからTime_code()で示されるTime_codeのデータエレメントが記述されていることを知る。つまり、extntion_and_user_data(i)のiが2のときは、このユーザデータがピクチャレイヤにあることを意味し、つまり、図26に示すように、このTime_code()241によって表されるデータエレメントは、ピクチャレイヤのユーザーで^タエリアに記述されているデータであるということである。

Timecode1のData_IDは、図15に示すように、”03”を表すビット列であり、Time_code1のデータは、画像の垂直プランギング期間に挿入されたタイムコードを示す、VITC (Vertical Interval Time Code) である。Time_code2のData_IDは、図15に示すように、”04”を表すビット列であり、Timecode2のデータは、記録媒体のタイムコードトラックに記録されたタイムコードを示す

、LTC(Longitudinal Time Code又はLinear Time Code)である。

図18及び図19は、Timecode()のシンタックスを示している図である。図18に示されているように、タイムコードは、72ビットのデータで表されており、図19具体的なデータ構造を示している。

図19において、color_frame_flagは、カラーフレーム情報に関するコントロールフラグを表わし、次のDrop_frame_flagは、ドロップフレームに関するコントロールフラグを表わしている。次の3ビット目から8ビット目までの6ビットは、タイムコードの‘フレーム’の部分を表わし、field_phaseは、フェイズコレクションに関するコントロールフラグを表わし、10ビット目から16ビット目までの7ビットは、タイムコードの‘秒’の部分を表わしている。17ビット目、34ビット目、51ビット目及び68ビット目の‘1’は、前述したように0が23個連続しないようにするためのマーカービットであって、このように所定間隔でマーカービットを挿入することによって、スタートコードエミュレーションを防止することができる。

18ビット目、26ビット目及び27ビット目のbinary_groupは、バイナリグループに関するコントロールフラグを表わし、19ビット目から25ビット目の7ビットは、タイムコードの‘分’の部分を表わし、28ビット目から33ビット目の6ビットは、タイムコードの‘時’の部分を表わしている。図14のElseif文において、nextbits()は、Picture Orderを示すビット列を検出すると、そのビット列の次ビットからPicture_Order()で示されるPictureOrderのデータメントが記述されていることを知る。ここで、Picture_Order()のData_IDは、図15に示すように、”05”を表すビット列である。実際に、エンコーダでエレメンタリストリーム(ES)に挿入するPicture_Order()のシンタックスを、図20を参照して説明する。まず、Data_IDは前述したように、そのData_ID以降のデータがPicture_

Orderのデータであることを示す8ビットのデータであり、その値は”05”である。DTS_presenceは、符号化順序DTS_counterの有無を表す1ビットのデータである。例えば、BピクチャのようにDTS_counter=PTS_counterとなる場合、表示順序PTS_counterのみが存在し、DTS_presenceのビットは”0”となる。逆に、PピクチャおよびIピクチャの場合、符号化順序DTS_counterと表示順序PTS_counterは同一ではないので、表示順序PTS_counterと符号化順序DTS_counterの双方が存在し、DTS_presenceのビットは1となる。尚、Picture_Order()に関するデータエレメントは、図26に示したように、Time_Code()と同じように、ピクチャレイヤのユーザーデータエリアに記述されている。PTS_counterは、前述したように、MPEGエンコーダ内のフィールドカウンタ102によって生成される値であって、入力ビデオデータにおいて1フィールドが入力されるごとにカウントアップを行う、表示順序を表す7ビットのデータである。この7ビットのデータは、0から127までの値をとるモジュロである。if文以降は、DTS_presenceのビットが1のとき、すなわち、PピクチャおよびIピクチャのとき、DTS_counterのカウントアップが実行される。Marker_bitsは、user_dataの記述されたビット列が、偶然に前述したスタートコードと一致し、画像破綻を引き起こす可能正が高い、スタートコードエミュレーションを防止するために、16ビットごとに挿入されるビットである。DTS_counterは、MPEGエンコーダ内のフィールドカウンタ102によって生成される値で、1フィールド分の符号化画像データが符号化されるごとにカウントアップを行う、符号化順序を表す7ビットのデータである。この7ビットのデータは、0から127までの値をとるモジュロである。図14に戻って、その次に記述されているwhile構文も、内容は前述した場合と同様であるので、ここではその説明を省略する。while構文が真のとき、次のif文において、nextbits()は、Ancillarydata

`ta`を示すビット列を検出すると、そのビット列の次ビットから`Ancillary_data()`で示される`Ancillary_data`のデータエレメントが記述されていることを知る。`Ancillary_data()`の`Data_ID`は、図15に示すように、”07”を表すビット列である。尚、`Ancillary_data()`に関するデータエレメントは、図26に示したように、`Picture_Order()`や`Time_Code()`と同じように、ピクチャレイヤのユーザーデータエリアに記述されている。この補助データに識別子を付加する`ancillary_data`のシンタックスを図21を参照して説明する。`Ancillary_data()`はピクチャレイヤの`user_data`として挿入され、データとしては`Field`識別子(`Field_ID`)、ラインの番号(`Line_number`)およびアンシラリーデータ(`ancillary_data`)が挿入される。`Data_ID`は、`user_data`領域において、`ancillary_data`であることを示す8ビットのデータであり、その値は図15に示したように”07”である。`Field_ID`は2ビットのデータであり、入力ビデオデータがプログレッシブビデオであるか否かを示すフラグである`progressive_sequence_flag`の値が”0”的き、つまり、入力ビデオがインターレースのビデオデータであるとき、符号化フレーム内のフィールドごとに`Field_ID`が付加される。

この`Field_ID`について、図7を参照して説明する。

`repeat_first_field`に”0”が設定されているとき、この符号化フレームにはフィールドが2枚存在し、`Field_ID`は、最初のフィールドに”0”、およびその次のフィールドに”1”が設定される。`repeat_first_field`に”1”が設定されているとき、この符号化フレームにはフィールドが3枚存在し、`Field_ID`としては、最初のフィールドに”0”が設定され、それ以降のフィールドに”1”, ”2”が設定される。図7Cを参照して、さらに詳しく説明する。図7Cの符号化ストリームは、図7Bに示される入力ビデオデータを符号化したときの、符号化ストリームに関する図で

ある。この符号化ストリームは、複数のアクセスユニット（AU1、AU2...）からなるエレメンタリーストリームから構成されており、図7Cは、そのエレメンタリーストリーム中にアンシラリーデータとField_IDに関する情報が記述されていることを示している図である。

この符号化ストリームは、フレームF1に関しては、Field_IDとして0、1及び2が設定されている。つまり、Field_ID=0のときに、そのフィールドに関連するアンシラリーデータ「0」がストリーム中に記述され、Field_ID=1のときに、そのフィールドに関連するアンシラリーデータ「1」がストリーム中に記述され、Field_ID=2のときに、そのフィールドに関連するアンシラリーデータ「2」がストリーム中に記述される。つまり、フレームF1に関するピクチャにおいて、図26に示すように、Ancillary_data()243に関するデータエレメントは、フレームF1内のフィールド数だけ繰り返されることを意味している。

Field_IDは、progressive_sequence_flagの値が”1”のとき、つまり入力ビデオデータが1であるとき、符号化フレームごとに付加される。Field_IDには、repeat_first_fieldとTop_field_firstにともに”0”が設定されているとき、その符号化フレームは1枚のprogressive_frameが存在するので、値”0”が設定され、repeat_first_fieldに値”1”およびTop_field_firstに値”0”が設定されているとき、その符号化フレームは2枚のprogressive_frameが存在するので、値”0”,”1”が設定され、repeat_first_fieldとTop_field_firstにともに”1”が設定されているとき、その符号化フレームは3枚のprogressive_frameが存在するので、値”0”乃至”2”が設定される。Line_numberは、14ビットのデータであり、各フレームにおけるアンシラリーデータが記述されている、ITU-R BT.656-3, SMPTE274M, SMPTE293M, SMPTE296Mで

規定されたライン番号を示す。Ancillary_data_lengthは、16ビットのデータであり、ancillary_data_payloadのデータ長を示す。Ancillary_data_payloadは、22ビットのデータからなる補助データの内容を表しており、Ancillary_data_payloadのAncillary_data_lengthの値がjの値（初期値0）より大きいとき、値j（Ancillary_data_lengthのデータ長）を1だけインクリメントして、そのjの値のビット列目から記述される。次のWhile構文は、bytealigned()のためのシンタックスを表しており、次のデータがbytealigned()でないとき（While構文が真のとき）、Zero_bit（1ビットのデータ”0”）を記述する。図14に戻って、次のElse_if文において、nextbits()は、History_dataを示すビット列を検出すると、そのビット列の次ビットからHistory_data()で示されるHistory_data()のData_IDは、図15に示すように、”08”を表すビット列であり、Data_IDが”08”で示されるデータは、符号化パラメータの履歴情報を含むHistory_dataを表している。このHistory_data()については、US特許出願09/265723に詳しく説明されているので、ここでは省略する。最後のif文において、nextbits()は、user_dataを示すビット列を検出すると、そのビット列の次ビットからuser_data()で示されるuser_dataのデータエレメントが記述されていることを知る。図14のnextbits()が、それぞれのデータエレメントが記述されていることを知るビット列は、図15に示すData_IDとして記述されている。ただし、Data_IDとして”00”を使用することは禁止されている。Data_IDが”80”で示されるデータは、制御フラグを表しており、Data_IDが”FF”で示されるデータは、user_dataを表している。図

22は、group_of_picture_header () のシンタックスを説明するための図である。このgroup_of_picture_header () によって定義されたデータエレメントは、group_start_code、time_code、closed_gop、およびbroken_linkから構成される。group_start_codeは、GOPレイヤの開始同期コードを示すデータである。time_codeは、GOPの先頭ピクチャのシーケンスの先頭からの時間を示すタイムコードである。closed_gopは、GOP内の画像が他のGOPから独立再生可能なことを示すフラグデータである。broken_linkは、編集などのためにGOP内の先頭のBピクチャが正確に再生できないことを示すフラグデータである。

extension_and_user_data (1) は、extension_and_user_data (0) と同じように、user_data () によって定義されるデータエレメントのみを記述するための関数である。次に、図23乃至図25を参照して、符号化ストリームのピクチャレイヤに関するデータエレメントを記述するためのpicture_header ()、picture_coding_extension ()、およびpicture_data ()について説明する。図23はpicture_header () のシンタックスを説明するための図である。このpicture_header () によって定義されたデータエレメントは、picture_start_code、temporal_reference、picture_coding_type、vbv_delay、full_pel_forward_vector、forward_f_code、full_pel_backward_vector、backward_f_code、extra_bit_picture、およびextra_information_pictureである。具体的には、picture_start_codeは、ピクチャレイヤの開始同期コードを表すデータである。temporal_referenceは、ピクチャの表示順を示す番号で、GOPの先頭でリセットされるデータである。picture

`_coding_type` は、ピクチャタイプを示すデータである。`vbv_delay` は、VBVバッファの初期状態を示すデータであって、各ピクチャ毎に設定されている。送信側システムから受信側システムに伝送された符号化エレメントリストリームのピクチャは、受信側システムに設けられたVBVバッファにバッファリングされ、`DTS (Decoding Time Stamp)` によって指定された時刻に、このVBVバッファから引き出され（読み出され）、デコーダに供給される。`vbv_delay` によって定義される時間は、復号化対象のピクチャがVBVバッファにバッファリングされ始めてから、符号化対象のピクチャがVBVバッファから引き出されるまでの時間、つまり `DTS` によって指定された時刻までの時間を意味する。このピクチャヘッダに格納された `vbv_delay` を使用することによって、VBVバッファのデータ占有量が不連続にならないシームレスなスプライシングが実現できる。`full_pel_forward_vector` は、順方向動きベクトルの精度が整数単位か半画素単位かを示すデータである。`forward_f_code` は、順方向動きベクトル探索範囲を表すデータである。`full_pel_backward_vector` は、逆方向動きベクトルの精度が整数単位か半画素単位かを示すデータである。`backward_f_code` は、逆方向動きベクトル探索範囲を表すデータである。`extra_bit_picture` は、後続する追加情報の存在を示すフラグである。この `extra_bit_picture` が「1」の場合には、次に `extra_information_picture` が存在し、`extra_bit_picture` が「0」の場合には、これに続くデータが無いことを示している。`extra_information_picture` は、規格において予約された情報である。図24は、`picture_coding_extension()` のシンタックスを説明するための図である。この `picture_coding_extension()` によって定義されたデータエレメントは、`extension_start_code`、`extension_start_code_identifier`、`f_code[0]`

[0]、f_code[0][1]、f_code[1][0]、f_code[1][1]、intra_dc_precision、picture_structure、top_field_first、frame_predictive_frame_dct、concealment_motion_vectors、q_scale_type、intra_vlc_format、alternate_scan、repeat_first_field、chroma_420_type、progressive_frame、composite_display_flag、v_axis、field_sequence、sub_carrier、burst_amplitude、およびsub_carrier_phaseから構成される。

extension_start_codeは、ピクチャレイヤのエクステンションデータのスタートを示す開始コードである。extension_start_code_identifierは、どの拡張データが送られるかを示すコードである。f_code[0][0]は、フォアード方向の水平動きベクトル探索範囲を表すデータである。f_code[0][1]は、フォアード方向の垂直動きベクトル探索範囲を表すデータである。f_code[1][0]は、バックワード方向の水平動きベクトル探索範囲を表すデータである。f_code[1][1]は、バックワード方向の垂直動きベクトル探索範囲を表すデータである。intra_dc_precisionは、DC係数の精度を表すデータである。picture_structureは、フレームストラクチャかフィールドストラクチャかを示すデータである。これは、フィールドストラクチャの場合は、上位フィールドか下位フィールドかもあわせて示す。top_field_firstは、フレームストラクチャの場合、最初のフィールドがトップフィールドであるのか、ボトムフィールドであるのかを示すフラグである。frame_predictive_frame_dctは、フレーム・ストラクチャの場合、フレーム・モードDCTの予測がフレーム・モードだけであることを示すデータである。concealment_motion_vectorsは、イン

トラマクロブロックに伝送エラーを隠蔽するための動きベクトルがついていることを示すデータである。q_scale_typeは、線形量子化スケールを利用するか、非線形量子化スケールを利用するかを示すデータである。intra_vlc_formatは、イントラマクロブロックに、別の2次元VLC(可変長符号)を使うかどうかを示すデータである。alternate_scanは、ジグザグスキャンを使うか、オルタネート・スキャンを使うかの選択を表すデータである。repeat_first_fieldは、復号化時にリピートフィールドを生成するか否かを示すフラグであって、復号化時の処理において、repeat_first_fieldが「1」の場合にはリピートフィールドを生成し、repeat_first_fieldが「0」の場合にはリピートフィールドを生成しないという処理が行われる。chroma_420_typeは、信号フォーマットが4:2:0の場合、次のprogressive_frameと同じ値、そうでない場合は0を表すデータである。progressive_frameは、そのピクチャが、順次走査できているかどうかを示すデータである。composite_display_flagは、ソース信号がコンポジット信号であったかどうかを示すデータである。v_axisは、ソース信号が、PALの場合に使われるデータである。field_sequenceは、ソース信号が、PALの場合に使われるデータである。sub_carrierは、ソース信号が、PALの場合に使われるデータである。burst_amplitudeは、ソース信号が、PALの場合に使われるデータである。sub_carrier_phaseは、ソース信号が、PALの場合に使われるデータである。図25は、picture_data()のシンタックスを説明するための図である。このpicture_data()によって定義されるデータエレメントは、slice()によって定義されるデータエレメントである。但し、ビットストリーム中に、slice()のスタートコードを示すslice_start_codeが存在しない場合には、このslice()によって定義されるデータエレメントはビットストリーム中に記述されていない。

slice () は、スライスレイヤに関するデータエレメントを記述するための関数であって、具体的には、slice_start_code、slice_quantiser_scale_code、intra_slice_flag、intra_slice、reserved_bits、extra_bit_slice、extra_information_slice、およびextra_bit_slice等のデータエレメントと、macroblock () によって定義されるデータエレメントを記述するための関数である。

slice_start_codeは、slice () によって定義されるデータエレメントのスタートを示すスタートコードである。slice_quantiser_scale_codeは、このスライスレイヤに存在するマクロブロックに対して設定された量子化ステップサイズを示すデータである。しかし、各マクロブロック毎に、quantiser_scale_codeが設定されている場合には、各マクロブロックに対して設定されたmacroblock_quantiser_scale_codeのデータが優先して使用される。intra_slice_flagは、ビットストリーム中にintra_sliceおよびreserved_bitsが存在するか否かを示すフラグである。intra_sliceは、スライスレイヤ中にノンイントラマクロブロックが存在するか否かを示すデータである。スライスレイヤにおけるマクロブロックのいずれかがノンイントラマクロブロックである場合には、intra_sliceは「0」となり、スライスレイヤにおけるマクロブロックの全てがノンイントラマクロブロックである場合には、intra_sliceは「1」となる。reserved_bitsは、7ビットのデータであって「0」の値を取る。extra_bit_sliceは、符号化ストリームとして追加の情報が存在することを示すフラグであって、次にextra_information_sliceが存在する場合には「1」に設定される。追加の情報が存在しない場合には「0」に設定される。macroblock () は、マクロブロックレイヤに

関するデータエレメントを記述するための関数であって、具体的には、macroblock_escape、macroblock_address_increment、およびmacroblock_quantiser_scale_code等のデータエレメントと、macroblock_modes()、およびmacroblock_vectors(s)によって定義されたデータエレメントを記述するための関数である。macroblock_escapeは、参照マクロブロックと前のマクロブロックとの水平方向の差が34以上であるか否かを示す固定ビット列である。参照マクロブロックと前のマクロブロックとの水平方向の差が34以上の場合には、macroblock_address_incrementの値に33をプラスする。macroblock_address_incrementは、参照マクロブロックと前のマクロブロックとの水平方向の差を示すデータである。もし、このmacroblock_address_incrementの前にmacroblock_escapeが1つ存在するのであれば、このmacroblock_address_incrementの値に33をプラスした値が、実際の参照マクロブロックと前のマクロブロックとの水平方向の差分を示すデータとなる。macroblock_quantiser_scale_codeは、各マクロブロック毎に設定された量子化ステップサイズである。各スライスレイヤには、スライスレイヤの量子化ステップサイズを示すslice_quantiser_scale_codeが設定されているが、参照マクロブロックに対してmacroblock_quantiser_scale_codeが設定されている場合には、この量子化ステップサイズを選択する。

次に、図27を参照して、マルチプレクサ162Aについて説明する。

マルチプレクサ162Aは、複数のパケットタイザ301-309と、複数のトランスポートストリーム生成回路(TS Gen.)311-319と、複数のシステムターゲットデコーダバッファ(STD Buffer)321-3

29と、多重化回路330と、多重化コントローラ300とを備えている。

パケットタイザ301-309は、MPEGエンコーダからそれぞれ出力されたエレメンタリーストリームを受け取り、エレメンタリーストリームをパケット化することによってパケット化エレメンタリーストリーム（PES）を生成する。

図28は、エレメンタリーストリーム（ES）、パケット化エレメンタリーストリーム（PES）、そして、トランスポートストリームパケットの関連を説明するための図である。

ソースビデオデータを符号化すると、例えば図28Bに示すような、各アクセスユニットAU1、AU2、……から構成されるエレメンタリーストリームが形成される。図28Cは、パケットタイザのパケット化処理について説明している図であって、パケットタイザは、複数のアクセスユニットをパッキングし、そのパケットの先頭部分にPESヘッダを付加する。

図29は、このPESヘッダを説明するための図面である。図29に示すように、PESヘッダは、パケット開始コード、ストリームID、パケット長、記号「10」、フラグ制御コード、PESヘッダ長、コンディショナルコーディングから構成される。MPEG規格では、このコンディショナルコーディングに、プレゼンテーションタイムスタンプ（PTS）及びデコーディングタイムスタンプ（DTS）の情報を入れないと決められている。トランスポートストリーム生成回路（TS_Gen.）311-319は、パケットタイザ301-309から出力されたパケット化エレメンタリーストリームから、図28Dに示すように、188バイトのトランスポートストリームパケットからなるトランスポートストリームをそれぞれ生成する。

システムターゲットデコーダバッファ（STD Buffer）321-329は、トランスポートストリーム生成回路311-319から出力されたトランスポートストリームを受け取り、バッファリングする。このSTDバッファは、MPEGデコーダ側の受信バッファがオーバーフロー及びアンダーフローしな

いようにシミュレーションを目的として設けられたバッファであって、MPEG規格によって決まっている固定容量のバッファである。多重化回路330は、システムターゲットデコーダバッファ321-329から、トランスポートストリームをそれぞれ受け取り、コントローラ300によって設定されたスケージューリングに従って多重化処理を実行する。

次に、各パケットタイマーの詳細な構成及び処理について、図27及び図30を参照して説明する。

各パケットタイマーは、受け取ったエレメンタリストリームをバッファリングするためのバッファ341と、その受け取ったエレメンタリストリームのシンタックスをページング（解析）するためのページング回路342と、バッファから出力されたエレメンタリーストリームをパケット化するパケット化回路343を備えている。

ページング回路342は、エレメンタリーストリームから、そのエレメンタリーストリーム中に記述されたPTS _counter 及びDTS _counter を抽出し、そのPTS _counter 及びDTS _counter をパケット化回路343に供給する。具体的には、ページング回路342は、受け取ったエレメンタリーストリームを可変長符号化し、ストリーム中に含まれているスタートコードなどの特殊なデータエレメントを検索する。このページング処理は、PTS _counter 及びDTS _counter を抽出することが目的であるので、シーケンスレイヤはGOPレイヤのスタートコードは無視し、まず、ピクチャレイヤのスタートコードを検索する。次に、32ビットのuser_data_start _code をストリーム中から見つけることによってピクチャレイヤのユーザデータエリアを見つけることができる。次に、このユーザデータエリアにおいて、Picture _order () に関するデータエレメントを見つけるために、“05” Data_IDを探す。このPicture _order () 関数の10ビット目から16ビット目及び17ビット目から23ビット目には、PTS _counter

r と DTS_counter が夫々記述されているので、ページング回路 342、このPTS_counter と DTS_counter とを抜き出して、パケット化回路 343 に供給する。パケット化回路 343 は、ページング回路 342 から PTS_counter 及び DTS_counter を受け取り、このPTS_counter 及び DTS_counter に関する情報に基づいて PTS と DTS を新たに生成する。本実施例では、PTS の値として PTS_counter の値そのものを使用し、DTS の値として、DTS_counter の値そのものを使用するようにしている。

図 30 は、本実施例のパケタイザを使用した場合において発生する最小遅延について説明するための図である。図 30 A は、入力ビデオデータを表し、図 30 B は、その入力ビデオデータを符号化した際のエレメンタリーストリームを表しており、図 30 C は、本実施例のMPEG エンコーダによって生成した符号化ストリームを使用し、本実施例のパケッタイザを使用した際の、パケッタイズドエレメンタリーストリームを表している。この図 30 A と図 30 B については、図 2 A 及び図 2 B と同じである。しかしながら、図 2 C と図 30 C を比較すると理解できるように、従来は、PTS を決定するパケッタイズドエレメンタリーストリームを生成する、つまり、PTS を決定するためには、I ピクチャを P ピクチャの間に存在する B ピクチャの数を N とすると $(N + 2)$ フレーム分の遅延が発生していたという問題については説明済みである。

本実施例のエンコーディング方法及びパケット化方法によれば、符号化ストリームから PTS を決定する遅延を 1 フレーム時間に押さえることができる。さらに、I ピクチャを P ピクチャの間に存在する B ピクチャの数に限らず、I ピクチャを P ピクチャの間に何枚もの B ピクチャが存在していたとしても、1 フレームの最小遅延となる。また、図 27 のような 9 チャンネル用のパケッタイザを設計しようとしたときに、9 個のフレームメモリで実現できるという、極めて大きな効果がある。

次に、図31を参照して、MPEGデコーダ144A-144Dについて説明する。各MPEGデコーダは、受信バッファ401と、可変長復号化回路402と、逆量子化回路403と、逆DCT回路404と、コントローラ405と、演算回路411と、動き補償回路412と、メモリ413及び414と、送信バッファ415と、バースバンドビデオ生成回路416と、多重化回路417とを備えている。

可変長復号回路402は、受信バッファからエレメンタリーストリームを受け取り、そのエレメンタリーストリームに対して可変長復号化処理を行うことによって所定のデータ長を有したデータエレメントからなるストリームを生成する。さらに、可変長復号回路402は、可変長復号化されたデータストリームのシンタックスをパージング（解析）することによって、ストリーム中から全てのコーディングパラメータを抽出し、コントローラ405に供給する。ストリーム中に重畠することがMPEG規格によって義務付けられているコーディングパラメータの例としては、ピクチャタイプ、動きベクトル、予測モード、DCTモード、量子化スケールコード、量子化テーブル情報等であって、基本的には、この符号化ストリームを生成する符号化処理において発生したパラメータである。

本実施例のユニークな点は、可変長符号化回路402は、上述したMPEG規格によって定められているコーディングパラメータだけではなく、エレメンタリーストリームのユーザーデータエリアに `MPEG_ES_Editing_information()` として記述されている情報を抽出することが、本実施例の特徴である。具体的には、符号化ストリーム中のシーケンスレイヤのユーザデータエリアには、`V-phase()` `H-phase()` に関する情報が `MPEG_ES_Editing_information()` として記述されており、符号化ストリーム中のピクチャレイヤのユーザデータエリアには、`Time_code()`、`Picture_order()`、`Ancillary_data()`、`History_data()` に関する情

報が `MPEG_ES_Editing_information()` として記述されており、可変長符号化回路402は、ストリーム中から、これらの `V-phase()` `H-phase()`、`Time_code()`、`Picture_order()`、`Ancillary_data()` 及び `History_data()` に関する情報を抽出し、コントローラ405に供給する。

逆量子化回路403は、可変長復号化回路402より供給された可変長復号化されたDCT係数データを、同じく可変長復号化回路402から供給された量子化スケールに基づいて逆量子化し、逆DCT回路404に出力する。

逆DCT回路404は、逆量子化回路403から供給された量子化DCT係数に対して、逆離散コサイン変換処理を施し、逆DCTされた画像データとして、演算器411に供給される。逆DCT回路404より演算器411に供給された画像データが、Iピクチャのデータである場合、そのデータは演算器411より出力され、演算器411の後に入力される画像データ（Pピクチャ又はBピクチャのデータ）の予測画像データ生成のために、フレームメモリの前方予測画像部414に供給されて記憶される。逆DCT回路404より供給された画像データが、その1フレーム前の画像データを予測画像データとするPピクチャのデータであり、前方予測モードのデータである場合、フレームメモリの前方予測画像部414に記憶されている、1フレーム前の画像データ（Iピクチャのデータ）が読み出され、動き補償回路412で可変長復号化回路402より出力された動きベクトルに対応する動き補償が施される。そして、演算器411において、逆DCT回路404より供給された画像データ（差分のデータ）と加算され、出力される。この加算されたデータ、すなわち、復号されたPピクチャのデータは、演算器411の後に入力される画像データ（Bピクチャ又はPピクチャのデータ）の予測画像データ生成のために、フレームメモリの後方予測画像部413に供給されて記憶される。Pピクチャのデータであっても、画像内予測モードのデータは、Iピクチャのデータと同様に、演算器411において処理は

行われず、そのまま後方予測画像部 413 に記憶される。この P ピクチャは、次の B ピクチャの次に表示されるべき画像であるため、この時点では、まだフォーマット変換回路 32 へ出力されない（上述したように、B ピクチャの後に入力された P ピクチャが、B ピクチャより先に処理され、伝送されている）。逆 DCT 回路 404 より供給された画像データが、B ピクチャのデータである場合、可変長復号化回路 402 より供給された予測モードに対応して、フレームメモリの前方予測画像部 414 に記憶されている I ピクチャの画像データ（前方予測モードの場合）、後方予測画像部 413 に記憶されている P ピクチャの画像データ（後方予測モードの場合）、又は、その両方の画像データ（両方向予測モードの場合）が読み出され、動き補償回路 412 において、可変長復号化回路 402 より出力された動きベクトルに対応する動き補償が施されて、予測画像が生成される。但し、動き補償を必要としない場合（画像内予測モードの場合）、予測画像は生成されない。このようにして、動き補償回路 412 で動き補償が施されたデータは、演算器 411 において、逆 DCT 回路 404 の出力と加算される。この加算出力は、バッファ 415 を介してベースバンドビデオ生成回路 416 に供給される。この送信バッファ 415 から出力されるビデオデータは、アクティブビデオエリアのみのビデオデータであってブランкиング期間などの補助的データは付与されていない。

コントローラ 405 は、可変長復号化回路 402 から供給されたピクチャタイプ、動きベクトル、予測モード、DCT モード、量子化スケールコード、量子化テーブル情報等のコーディングパラメータに関する情報にもとづいて、上述した各回路のオペレーションをコントロールする。

さらには、コントローラ 405 は、可変長符号化回路 402 から M P E G _ E S _ E d i t i n g _ i n f o r m a t i o n () として供給された V - p h a s e 及び H - p h a s e に関する情報に基づいて、ベースバンドビデオ生成回路 416 をコントロールする。符号化ストリーム中から抽出された V - p h a s e は、入力ビデオデータの全画素エリアにおけるアクティブ

ビデオエリアの垂直方向の位置を示しており、また、そのH-phase は、入力ビデオデータの全画素エリアにおけるアクティブビデオエリアの水平方向の位置を示している。よって、コントローラ405は、バッファ415から出力された復号化ビデオデータを、ブランкиングイメージを有した全画素エリア上に、V-phase 及び H-phase によって示される垂直及び水平位置にマッピングするようにベースバンドビデオ生成回路をコントロールする、または、V-phase 及び H-phase によって示される垂直及び水平位置に基づいて、アクティブエリアの復号ビデオデータと、全画素エリアのブランкиングイメージを合成するようにベースバンドビデオ生成回路をコントロールする。その結果、ベースバンドビデオ生成回路416から出力されるビデオデータは、MPEGエンコーダに供給された入力ビデオデータのブランкиング期間と全く同じブランкиング期間を有している。

コントローラは405は、符号化ストリームから抽出された Ancillary_data、Line_number、Field_ID、Time_code_1 及び Time_code_2 を供給する、又はField_ID に基づいて、Ancillary_data、Time_code_1 又はTime_code_2 の多重化回路417の多重化処理をコントロールする。具体的には、既に図7Cにおいて説明したように、Field_IDはフィールド毎に付与されたAncillary_dataと関連付けされているので、多重化回路417はこのField_IDによって同定されるフィールドのブランкиング期間に、Field_IDと関連付けられているAncillary_dataを重畳する。例えば、Field_IDが「2」の場合には、フレームにおいて3番目のフィールドであることがわかるので、符号化ストリーム中において、Field_IDが「2」に関連するアンシラリーデータとして伝送されてきたAncillary_dataを、Field_IDが「2」の3番目のフィールドのブランкиング期間に重畳する。多

重化回路417は、さらに、Ancillary_dataをブランкиング期間に重複する際、Ancillary_dataと一緒に伝送されているLine_numberによって指定されるライン位置に、そのAncillary_dataを重複する。

従って、多重化回路417から出力されたビデオデータは、MPEGエンコーダに供給された入力ビデオデータと全く同じブランкиング期間、全く同じ位置のアクティブビデオエリア、全く同じライン位置にかつ同じ内容のアンシラリデータを有している。

よって、本実施例によれば、ビデオデータを送信側システムから受信側システムに伝送する際に、MPEGエンコード及びデコード処理を行ったとしても、そのコーディング処理によって入力ビデオデータがそもそも有している固有情報や入力ビデオデータに付加されたアンシラリーデータが無くなることはない。

産業上の利用可能性

本発明は放送局等で頻繁にビデオデータをエンコード及びデコードする際に利用できる。

請求の範囲

1. 入力ビデオデータをエンコードするためのエンコーディング装置において、上記入力ビデオデータから、上記ビデオデータのプランギング期間に付加されているアンシラリーデータを抽出する抽出手段と、上記入力ビデオデータをエンコードし、符号化ストリームを生成する符号化手段と、上記符号化ストリーム中に上記アンシラリーデータが挿入されるように、上記符号化手段をコントロールするコントロール手段とを備えたことを特徴とするエンコーディング装置。
2. 上記コントロール手段は、上記アンシラリーデータを上記符号化ストリームのピクチャレイヤのユーザーデータエリアに記述することを特徴とする請求の範囲第1項に記載のエンコーディング装置。
3. 上記コントロール手段は、上記符号化ストリームのピクチャレイヤの `extension_and_user_data(i)` 中に、上記アンシラリーデータを記述することを特徴とする請求の範囲第1項に記載のエンコーディング装置。
3. 上記コントロール手段は、上記符号化ストリームのピクチャレイヤのユーザーデータエリアに、上記アンシラリーデータをフィールド単位又はフレーム単位に記述するように上記符号化手段をコントロールすることを特徴とする請求の範囲第1項に記載のエンコーディング装置。
4. 上記コントロール手段は、上記入力ビデオデータにおいて上記アンシラリーデータが挿入されていたフィールドを示すフィールドIDと、上記アンシラリー

データとを関連付けて上記ピクチャレイヤに記述するように上記符号化手段をコントロールすることを特徴とする請求の範囲第1項に記載のエンコーディング装置。

5. 上記コントロール手段は、上記入力ビデオデータにおいて上記アンシラリーデータが挿入されていたフィールドを示すフィールドIDと、上記アンシラリーデータが挿入されていたライン番号と、上記アンシラリーデータとを関連付けて上記ピクチャレイヤに記述するように上記符号化手段をコントロールすることを特徴とする請求の範囲第1項に記載のエンコーディング装置。

6. 上記入力ビデオデータの1フレームが複数のフィールドから構成され、上記コントロール手段は、上記符号化ストリームのピクチャレイヤに、各フィールド毎に上記アンシラリーデータを夫々記述することを特徴とする請求の範囲第1項に記載のエンコーディング装置。

7. 上記符号化ストリーム中に記述される上記アンシラリーデータは、可変長符号化されたデータストリームであることを特徴とする請求の範囲第1項に記載のエンコーディング装置。

8. 上記アンシラリーデータは、クローズドキャプションデータ及び／又はテレキストデータであることを特徴とする請求の範囲第1項に記載のエンコーディング装置。

9. 上記アンシラリーデータは、上記符号化ストリームのピクチャレイヤにおいて、`Ancillary_data()` によって表されるシンタックスに基づいて記述されることを特徴とする請求の範囲第1項に記載のエンコーディング装置。

10. 上記入力ビデオデータは、3-2 プルダウン処理された 30 Hz のビデオデータであって、

上記抽出手段と上記符号化手段との間に接続され、上記入力ビデオデータに対して、逆 3-2 プルダウン処理を行うことによって逆プルダウン処理されたビデオデータを生成するための逆プルダウン手段をさらに備えたことを特徴とする請求の範囲第 1 項に記載のエンコーディング装置。

11. 入力ビデオデータをエンコードするためのエンコーディング方法において、

上記入力ビデオデータから、上記ビデオデータのプランキング期間に付加されているアンシラリーデータを抽出するステップと、

上記入力ビデオデータをエンコードし、符号化ストリームを生成するステップと、

上記符号化ストリーム中に上記アンシラリーデータが挿入されるように、上記符号化手段をコントロールするステップと

を備えたことを特徴とするエンコーディング方法。

12. 入力ビデオデータをエンコードするためのエンコーディング装置において、

上記入力ビデオデータから、上記ビデオデータのプランキング期間に付加されているアンシラリーデータを抽出する手段と、

上記入力ビデオデータをエンコードし、符号化ストリームを生成する手段と、

上記符号化ストリームと一緒に上記アンシラリーデータが伝送されるように、

上記符号化ストリーム中に上記アンシラリーデータを挿入する手段と

を備えたことを特徴とするエンコーディング装置。

13. 入力ビデオデータをエンコードするためのエンコーディング方法において、

上記入力ビデオデータから、上記ビデオデータのブランкиング期間に付加されているアンシラリーデータを抽出するステップと、

上記入力ビデオデータをエンコードし、符号化ストリームを生成するステップと、

上記符号化ストリームと一緒に上記アンシラリーデータが伝送されるように、上記符号化ストリーム中に上記アンシラリーデータを挿入するステップとを備えたことを特徴とするエンコーディング方法。

14. 入力ビデオデータをエンコードするためのエンコーディング装置において

上記入力ビデオデータから、上記ビデオデータのブランкиング期間に付加されているアンシラリーデータを抽出する手段と、

上記入力ビデオデータのアクティブエリアをエンコードすることによって、符号化ストリームを生成する手段と、

上記アンシラリーデータを上記符号化ストリーム中に記述することによって、上記符号化ストリームと一緒に上記アンシラリーデータを伝送する手段とを備えたことを特徴とするエンコーディング装置。

15. 入力ビデオデータをエンコードするためのエンコーディング方法において

上記入力ビデオデータから、上記ビデオデータのブランкиング期間に付加されているアンシラリーデータを抽出するステップと、

上記入力ビデオデータのアクティブエリアをエンコードすることによって、符号化ストリームを生成するステップと、

上記アンシラリーデータを上記符号化ストリーム中に記述することによって、上記符号化ストリームと一緒に上記アンシラリーデータを伝送するステップと

を備えたことを特徴とするエンコーディング方法。

16. 入力ビデオデータをエンコードするためのエンコーディング装置において

、
上記入力ビデオデータが有して居る固有情報を上記入力ビデオデータから抽出する
抽出手段と、

上記入力ビデオデータをエンコードし、符号化ストリームを生成する符号化手
段と、上記符号化ストリームと一緒に上記記固有情報が伝送されるように、上記
符号化ストリーム中に上記固有情報を挿入する手段と

を備えたことを特徴とするエンコーディング装置。

17. 上記固有情報は、上記符号化ストリームのシーケンスレイヤのユーザデー
タエリヤに挿入されることを特徴とする請求の範囲第16項に記載のエンコーデ
ィング装置。

18. 上記固有情報は、上記入力ビデオデータにおけるアクティブビデオのスタ
ート位置を示す情報であつて、

上記符号化手段は、上記アクティブビデオのビデオデータを符号化することを
特徴とする請求の範囲第16項に記載のエンコーディング装置。

19. 上記固有情報は、上記符号化ストリームのシーケンスレイヤにおいて、V
_p h a s e () とH_p h a s e () によって表されるシンタックスに基づい
て記述されることを特徴とする請求の範囲第16項に記載のエンコーディング裝
置。

20. 入力ビデオデータをエンコードするためのエンコーディング方法において

上記入力ビデオデータが有している固有情報を上記入力ビデオデータから抽出するステップと、

上記入力ビデオデータをエンコードし、符号化ストリームを生成するステップと、

上記符号化ストリームと一緒に上記記固有情報が伝送されるように、上記符号化ストリーム中に上記固有情報を挿入するステップと
を備えたことを特徴とするエンコーディング方法。

21. 入力ビデオデータをエンコードするためのエンコーディング装置において

上記入力ビデオデータに対するアクティブビデオのフェーズを示すフェーズ情報を、上記入力ビデオデータから得る手段と、

上記入力ビデオデータをエンコードし、符号化ストリームを生成する手段と、
上記符号化ストリームと一緒に上記フェーズ情報が伝送されるように、上記符号化ストリームに上記フェーズ情報を付加する手段と
を備えたことを特徴とするエンコーディング装置。

22. 入力ビデオデータをエンコードするためのエンコーディング方法において

上記入力ビデオデータに対するアクティブビデオのフェーズを示すフェーズ情報を、上記入力ビデオデータから得るステップと、

上記入力ビデオデータをエンコードし、符号化ストリームを生成するステップと、
上記符号化ストリームと一緒に上記フェーズ情報が伝送されるように、上記符号化ストリームに上記フェーズ情報を付加するステップと
を備えたことを特徴とするエンコーディング方法。

23. 入力ビデオデータをエンコードするためのエンコーディング装置において

、
上記入力ビデオデータに対するアクティブビデオのフェーズを示すフェーズ情報、上記入力ビデオデータから得る手段と、
上記入力ビデオデータをエンコードし、符号化ストリームを生成する手段と、
上記符号化ストリームのシーケンスレイヤに上記フェーズ情報を記述する手段
と
を備えたことを特徴とするエンコーディング装置。

24. 入力ビデオデータをエンコードするためのエンコーディング方法において

、
上記入力ビデオデータに対するアクティブビデオのフェーズを示すフェーズ情報、上記入力ビデオデータから得るステップと、
上記入力ビデオデータをエンコードし、符号化ストリームを生成するステップ
と、
上記符号化ストリームのシーケンスレイヤに上記フェーズ情報を記述するステップと
を備えたことを特徴とするエンコーディング方法。

25. 入力ビデオデータをエンコードすることによって生成された符号化ストリームをデコードするためのデコーディング装置において、

上記符号化ストリームから、上記符号化ストリーム中に含まれていアンシラリーデータを抽出する抽出手段と、
上記符号化ストリームをデコードし、復号化ビデオデータを生成する復号化手段と、
上記復号化ビデオデータのブランкиング期間に、上記アンシラリーデータを多重化する多重化手段と

を備えたことを特徴とするデコーディング装置。

26. 上記抽出手段は、上記符号化ストリームのシンタックスをパージングすることによって、上記符号化ストリームのピクチャレイヤのユーザデータエリアから上記アンシラリーデータを抽出することを特徴とする請求の範囲第25項に記載のデコーディング装置。

27. 上記符号化ストリーム中には、上記入力ビデオデータにおいて上記アンシラリーデータが挿入されていたフィールドを示すフィールドIDが上記アンシラリーデータと関連付けて記述されており、

上記多重化手段は、上記フィールドIDによって指定されたフィールドに上記アンシラリーデータを多重化することを特徴とする請求の範囲第25項に記載のデコーディング装置。

28. 上記符号化ストリーム中には、上記入力ビデオデータにおいて上記アンシラリーデータが挿入されていたラインを示すライン番号が上記アンシラリーデータと関連付けて記述されており、

上記多重化手段は、上記ライン番号によって指定されるラインに、上記アンシラリーデータを多重化することを特徴とする請求の範囲第25項に記載のデコーディング装置。

29. 上記アンシラリーデータは、上記符号化ストリームのピクチャレイヤのA
n c i l l a r y _ d a t a () の中に記述されていることを特徴とする請求の範囲第25項に記載のデコーディング装置。

30. 入力ビデオデータをエンコードすることによって生成された符号化ストリームをデコードするためのデコーディング方法において、

上記符号化ストリームから、上記符号化ストリーム中に含まれていアンシラリーデータを抽出するステップと、

上記符号化ストリームをデコードし、復号化ビデオデータを生成するステップと、

上記復号化ビデオデータのプランキング期間に、上記アンシラリーデータを多重化するステップと

を備えたことを特徴とするデコーディング方法。

3 1. 入力ビデオデータを符号化することによって生成された符号化ストリームをデコードするためのデコーディング装置において、

上記符号化ストリームのシンタックスを解析することによって、上記符号化ストリーム中に含まれていアンシラリーデータを得る手段と、

上記符号化ストリームをデコードし、復号化ビデオデータを生成する復号化手段と、

上記入力ビデオデータと上記符号化ビデオデータとが同じアンシラリーデータを有するように、上記復号化ビデオデータに上記アンシラリーデータを多重化する手段と

を備えたことを特徴とするデコーディング装置。

3 2. 入力ビデオデータを符号化することによって生成された符号化ストリームをデコードするためのデコーディング方法において、

上記符号化ストリームのシンタックスを解析することによって、上記符号化ストリーム中に含まれていアンシラリーデータを得るステップと、

上記符号化ストリームをデコードし、復号化ビデオデータを生成するステップと、

上記入力ビデオデータと上記符号化ビデオデータとが同じアンシラリーデータを有するように、上記復号化ビデオデータに上記アンシラリーデータを多重化す

るステップと
を備えたことを特徴とするデコーディング方法。

3 3. 入力ビデオデータを符号化することによって生成された符号化ストリーム
をデコードするためのデコーディング装置において、

上記符号化ストリームのピクチャエリアに含まれていアンシラリーデータを、
上記符号化ストリームから得る手段と、

上記符号化ストリームをデコードし、復号化ビデオデータを生成する復号化手
段と、

上記復号化ビデオデータと上記アンシラリーデータとを多重化することによっ
て、上記入力ビデオデータと同じデータを生成する手段と
を備えたことを特徴とするデコーディング装置。

3 4. 入力ビデオデータを符号化することによって生成された符号化ストリーム
をデコードするためのデコーディング方法において、

上記符号化ストリームのピクチャエリアに含まれていアンシラリーデータを、
上記符号化ストリームから得る手段と、

上記符号化ストリームをデコードし、復号化ビデオデータを生成する手段と、
上記復号化ビデオデータと上記アンシラリーデータとを多重化することによっ
て、上記入力ビデオデータと同じデータを生成する手段と
を備えたことを特徴とするデコーディング方法。

3 5. 入力ビデオデータをエンコードすることによって生成された符号化ストリ
ームをデコードするためのデコーディング装置において、

上記符号化ストリームから、上記符号化ストリーム中に含まれている情報であ
って、上記入力ビデオデータの固有情報を抽出する抽出手段と、

上記符号化ストリームをデコードし、復号化ビデオデータを生成する復号化手段と、

上記固有情報に基づいて、上記復号化ビデオデータからベースバンドビデオデータを生成する生成手段と

を備えたことを特徴とするデコーディング装置。

3 6. 上記抽出手段は、上記符号化ストリームのシンタックスをパージングすることによって、上記符号化ストリームのシーケンスレイヤのユーザデータエリアから上記固有情報を抽出することを特徴とする請求の範囲第3 5項に記載のデコーディング装置。

3 7. 上記固有情報は、上記入力ビデオデータにおけるアクティブビデオの位置を示す情報であって、上記符号化ストリームのシーケンスレイヤにおいて、V_phase () とH_phase () によって表されるシンタックスに基づいて記述されていることを特徴とする請求の範囲第3 5項に記載のデコーディング装置。

3 8. 上記固有情報は、上記入力ビデオデータにおけるアクティブビデオの位置を示す情報であって、

上記生成手段は、上記固有情報に基づいて、上記入力ビデオデータにおけるアクティブビデオの位置と同じ位置に、上記復号化ビデオデータがマッピングされたベースバンドビデオデータを生成することを特徴とする請求の範囲第3 5項に記載のデコーディング装置。

3 9. 上記固有情報は、上記入力ビデオデータにおけるアクティブビデオの位置を示す情報であって、

上記生成手段は、上記固有情報に基づいて、上記復号化ビデオデータに対して

プランギング期間を付加することによってベースバンドビデオデータを生成することを特徴とする請求の範囲第35項に記載のデコーディング装置。

40. 入力ビデオデータをエンコードすることによって生成された符号化ストリームをデコードするためのデコーディング方法において、

上記符号化ストリームから、上記符号化ストリーム中に含まれている情報であって、上記入力ビデオデータの固有情報を抽出するステップと、

上記符号化ストリームをデコードし、復号化ビデオデータを生成するステップと、

上記固有情報に基づいて、上記復号化ビデオデータからベースバンドビデオデータを生成するステップと

を備えたことを特徴とするデコーディング方法。

41. 入力ビデオデータをエンコードすることによって生成された符号化ストリームをデコードするためのデコーディング装置において、

上記符号化ストリームから、上記符号化ストリーム中に含まれている情報であって、上記入力ビデオデータの固有情報を抽出する抽出手段と、

上記符号化ストリームをデコードし、復号化ビデオデータを生成する復号化手段と、

上記固有情報に基づいて、上記復号化ビデオデータを処理する手段と
を備えたことを特徴とするデコーディング装置。

42. 入力ビデオデータをエンコードすることによって生成された符号化ストリームをデコードするためのデコーディング方法において、

上記符号化ストリームから、上記符号化ストリーム中に含まれている情報であって、上記入力ビデオデータの固有情報を抽出するステップと、

上記符号化ストリームをデコードし、復号化ビデオデータを生成するステップ

と、

上記固有情報に基づいて、上記復号化ビデオデータを処理するステップと
を備えたことを特徴とするデコーディング方法。

4 3. 入力ビデオデータをエンコードすることによって生成された符号化ストリームをデコードするためのデコーディング装置において、

上記符号化ストリームから、上記符号化ストリーム中に含まれている情報であ
って、上記入力ビデオデータの固有情報を抽出する抽出手段と、

上記符号化ストリームをデコードし、復号化ビデオデータを生成する復号化手
段と、上記固有情報に基づいて、上記復号化ビデオデータを処理する手段と
を備えたことを特徴とするデコーディング装置。

4 4. 入力ビデオデータをエンコードすることによって生成された符号化ストリームをデコードするためのデコーディング方法において、

上記符号化ストリームから、上記符号化ストリーム中に含まれている情報であ
って、上記入力ビデオデータの固有情報を抽出するステップと、

上記符号化ストリームをデコードし、復号化ビデオデータを生成するステップと
と、

上記固有情報に基づいて、上記復号化ビデオデータを処理するステップと
を備えたことを特徴とするデコーディング方法。

4 5. 入力ビデオデータを符号化するエンコーディング手段と、上記符号化手段
によって符号化された符号化ストリームを受け取って、上記符号化ストリームを
復号化し復号化ビデオデータを生成するデコーディング手段とを備えたコーディ
ングシステムにおいて、

上記エンコーディング手段は、

上記符号化ストリームを生成するために上記入力ビデオデータを符号化する手

段と、

上記入力ビデオデータに付与されているアンシラリーデータを上記符号化ストリーム中に挿入する手段とを備え、

上記デコーディング手段は、

上記復号化ビデオデータを生成するために上記符号化ストリームを復号化する手段と、

上記符号化ストリームと共に伝送されてきたアンシラリーデータを、上記復号化ビデオデータに多重化する手段とを備えていることを特徴とするコーディングシステム。

4 6. 入力ビデオデータを符号化するエンコーディングプロセスと、上記エンコーディングプロセスによって符号化された符号化ストリームを受け取って、上記符号化ストリームを復号化するデコーディングプロセスとを行うコーディングシステムを使用して、上記入力ビデオデータに対して符号化プロセス及び復号化プロセスを行うためのコーディング方法において、

上記エンコーディングプロセスは、

上記符号化ストリームを生成するために上記入力ビデオデータを符号化するステップと、上記入力ビデオデータに関連するアンシラリーデータを上記符号化ストリーム中に挿入するステップとを含み、

上記デコーディングプロセスは、

上記復号化ビデオデータを生成するために上記符号化ストリームを復号化するステップと、上記符号化ストリームと共に伝送されてきたアンシラリーデータを、上記復号化ビデオデータに多重化するステップとを含んでいることを特徴とするコーディング方法。

4 7. 入力ビデオデータを符号化するエンコーディング手段と、上記符号化手段によって符号化された符号化ストリームを受け取って、上記符号化ストリームを

復号化し復号化ビデオデータを生成するデコーディング手段とを備えたコーディングシステムにおいて、

上記エンコーディング手段は、

上記符号化ストリームを生成するために上記入力ビデオデータを符号化する手段と、上記入力ビデオデータが有して居る固有情報を上記符号化ストリーム中に挿入する手段とを備え、

上記デコーディング手段は、

上記復号化ビデオデータを生成するために上記符号化ストリームを復号化する手段と、上記符号化ストリームと共に伝送されて居た固有情報に基づいて、上記復号化ビデオデータを処理する手段とを備えたことを特徴とするコーディングシステム。

4.8. 入力ビデオデータを符号化するエンコーディングプロセスと、上記エンコーディングプロセスによって符号化された符号化ストリームを受け取って、上記符号化ストリームを復号化するデコーディングプロセスとを行うコーディングシステムを使用して、上記入力ビデオデータに対して符号化プロセス及び復号化プロセスを行うためのコーディング方法において、

上記エンコーディングプロセスは、

上記符号化ストリームを生成するために上記入力ビデオデータを符号化するステップと、上記入力ビデオデータに関連する固有情報を上記符号化ストリーム中に挿入するステップとを含み、

上記デコーディングプロセスは、

上記復号化ビデオデータを生成するために上記符号化ストリームを復号化するステップと、上記符号化ストリームと共に伝送されて居た固有情報に基づいて、上記復号化ビデオデータを処理するステップとを含んでいることを特徴とするコーディング方法。

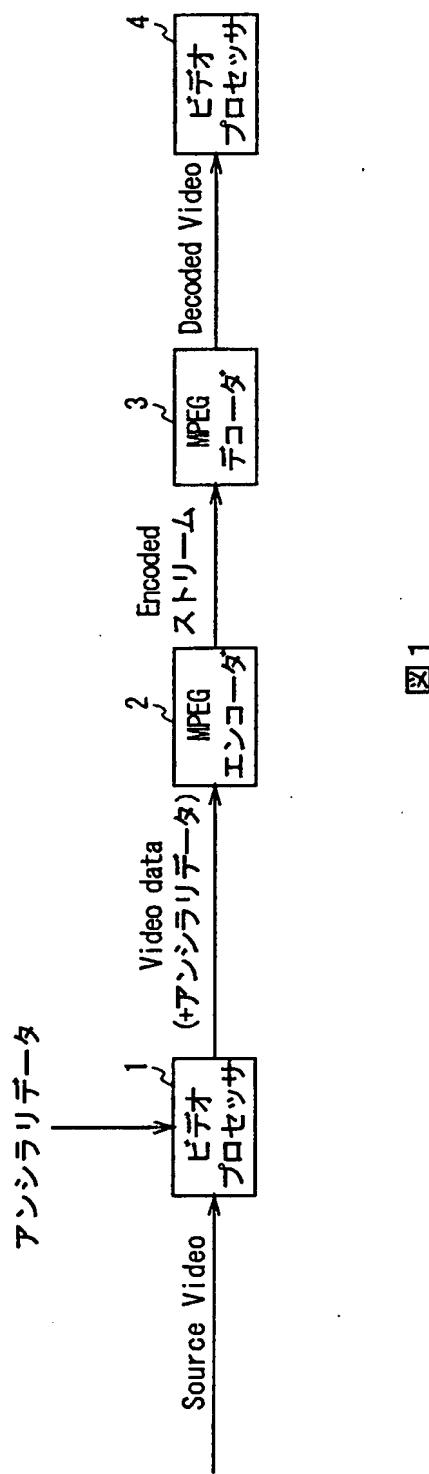


図 1

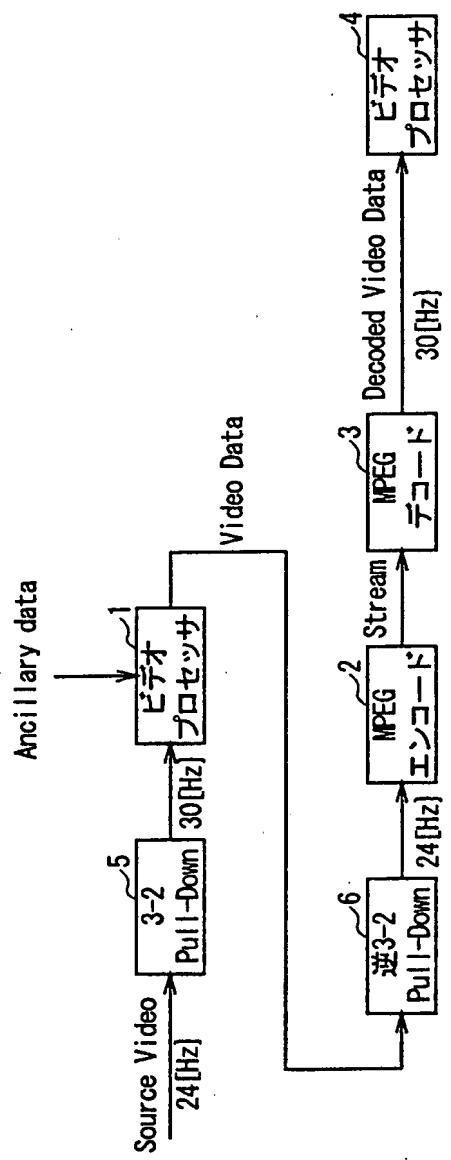
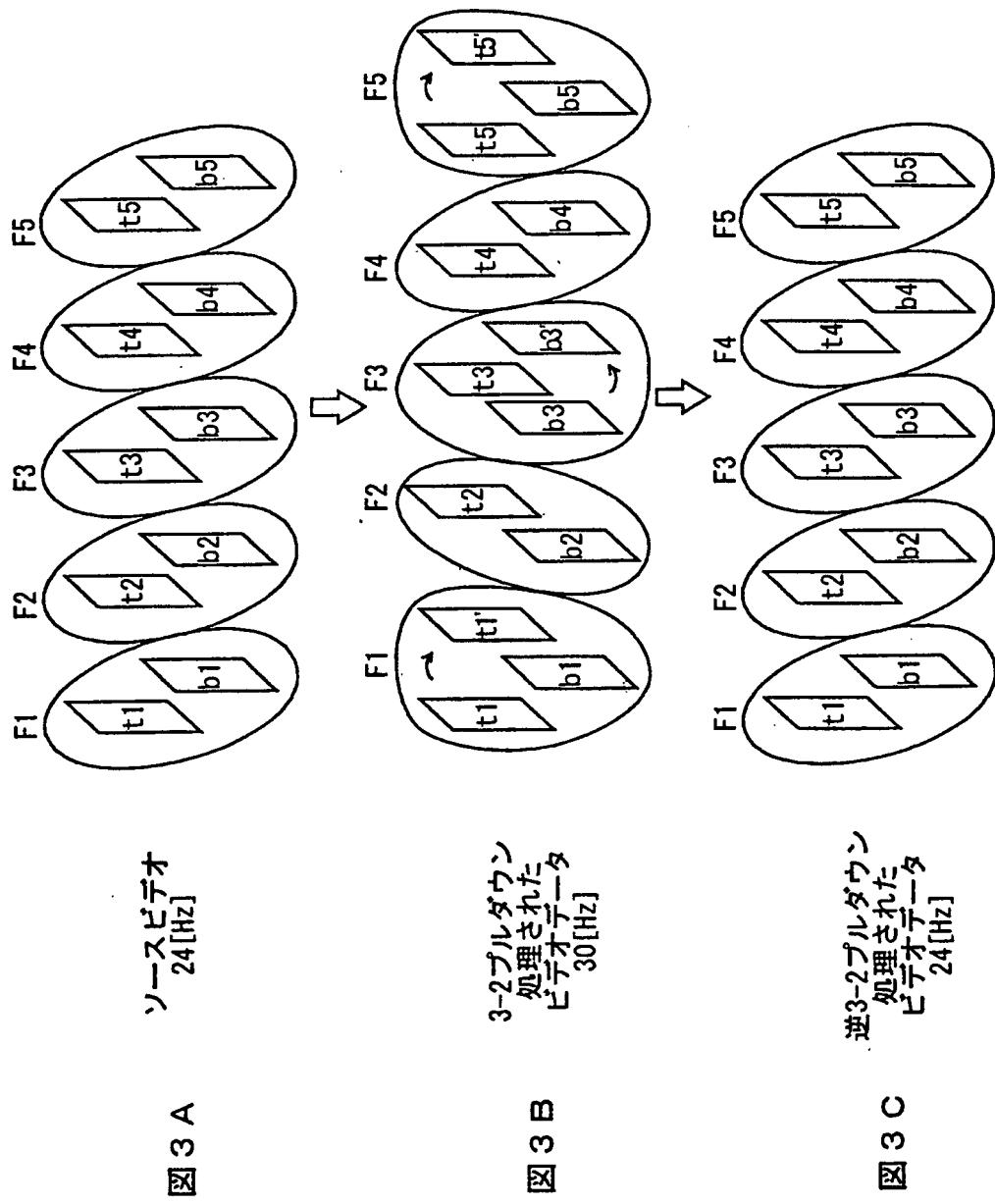
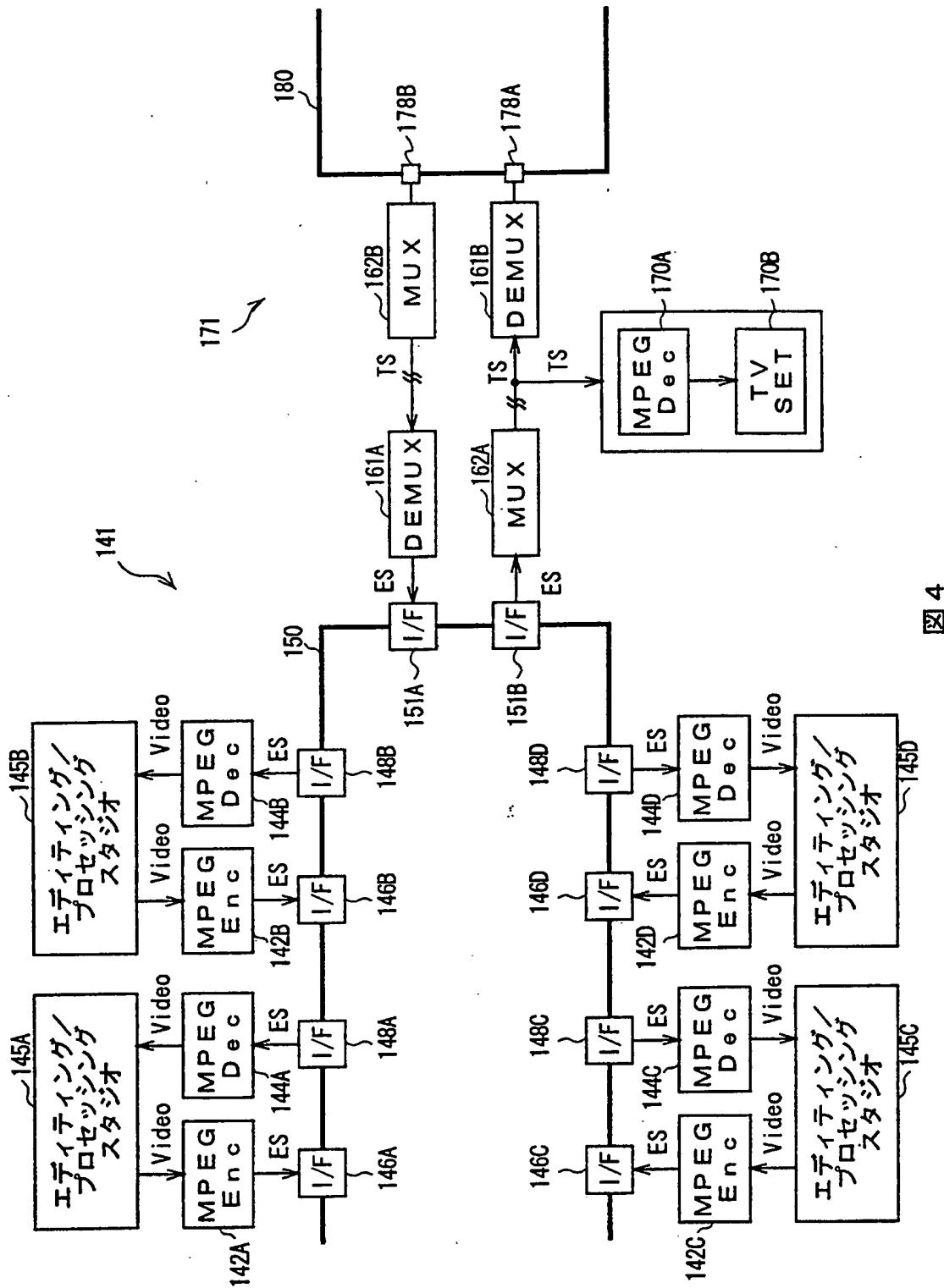
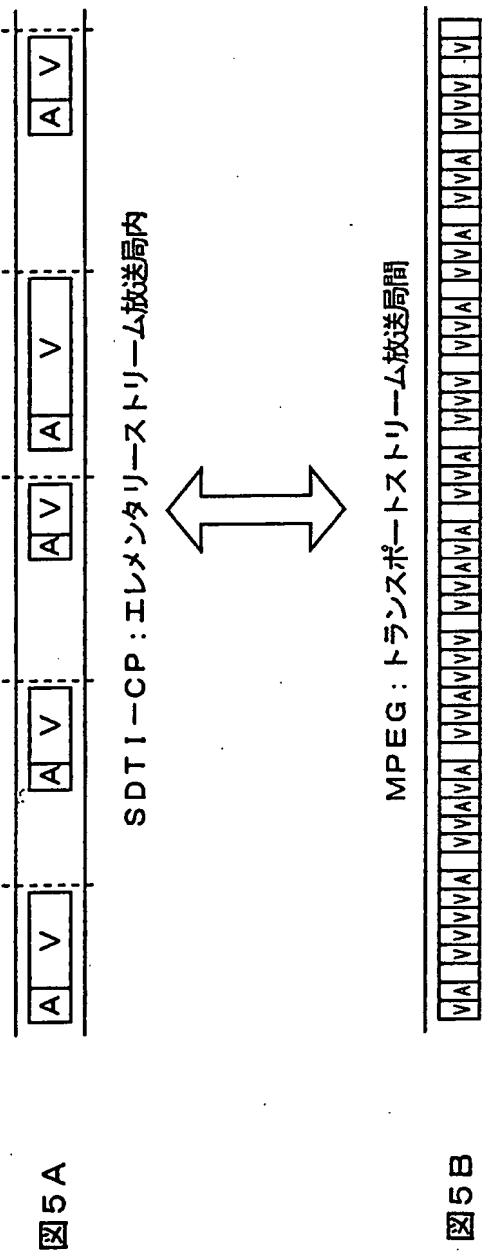


図2





4



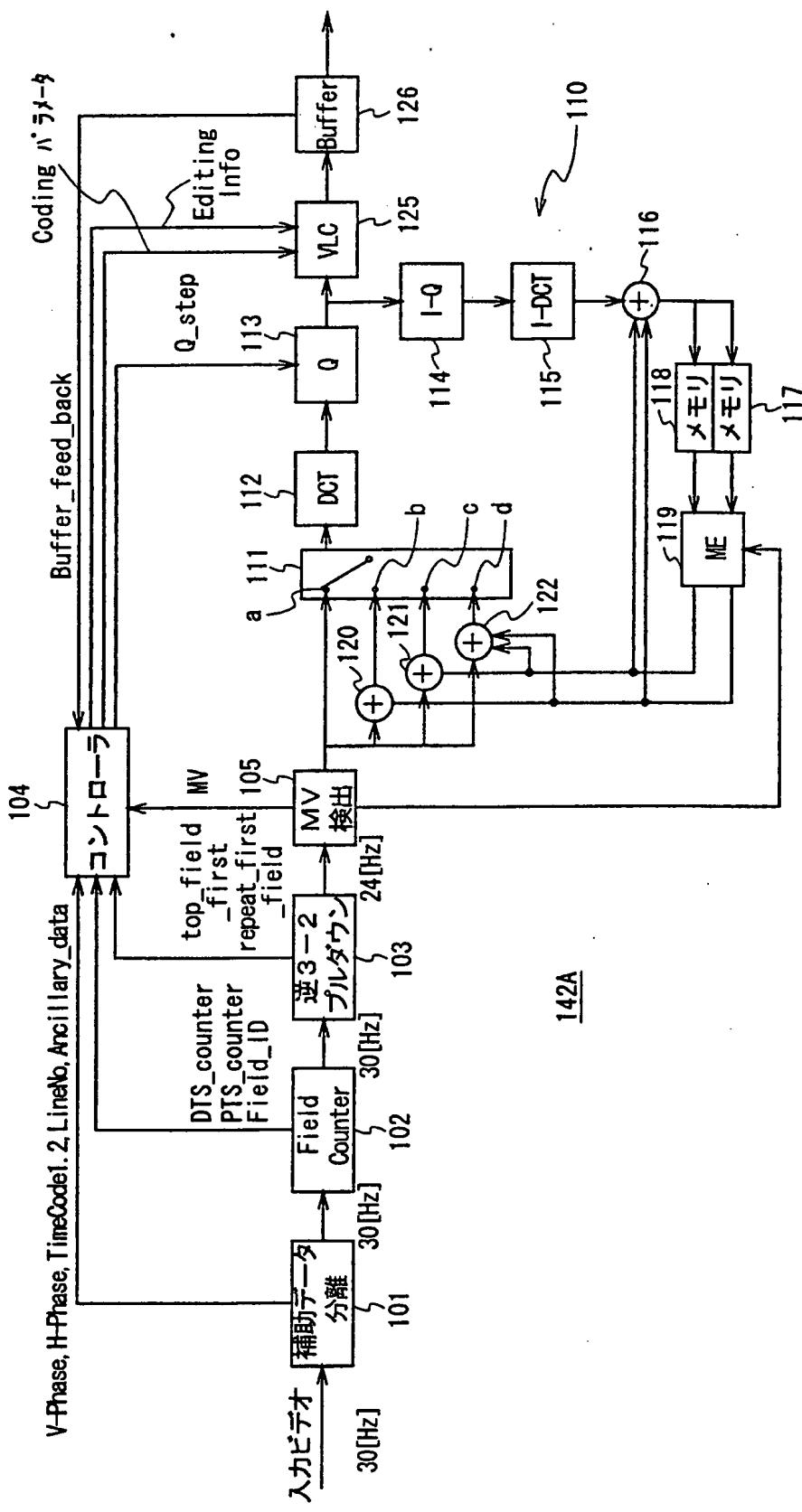
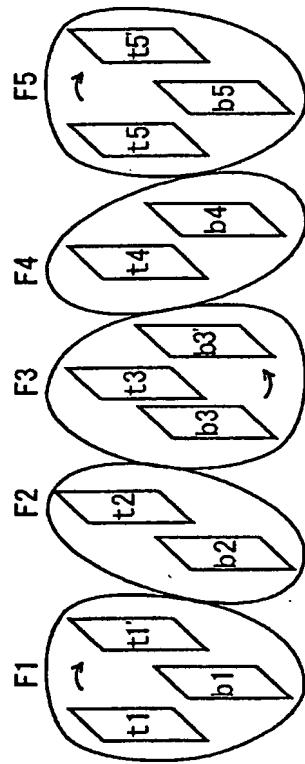
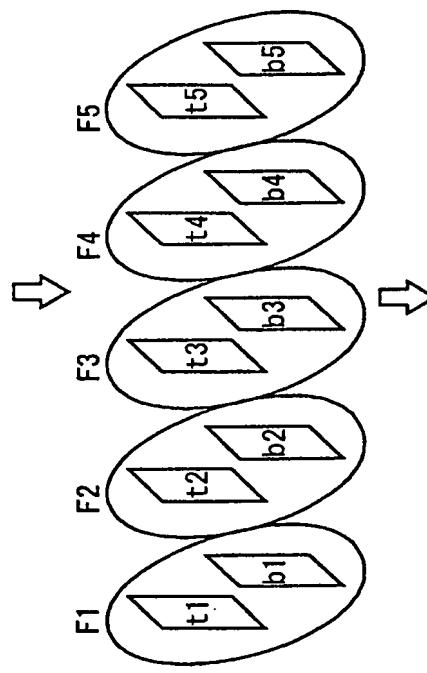


図 6



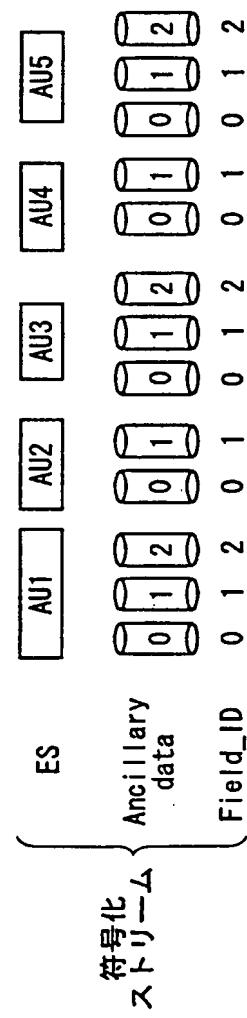
3-2 プルダウ
ン
処理された
ビデオデータ
30[Hz]

図 7 A



逆3-2 プルダウ
ン
処理された
ビデオデータ

図 7 B



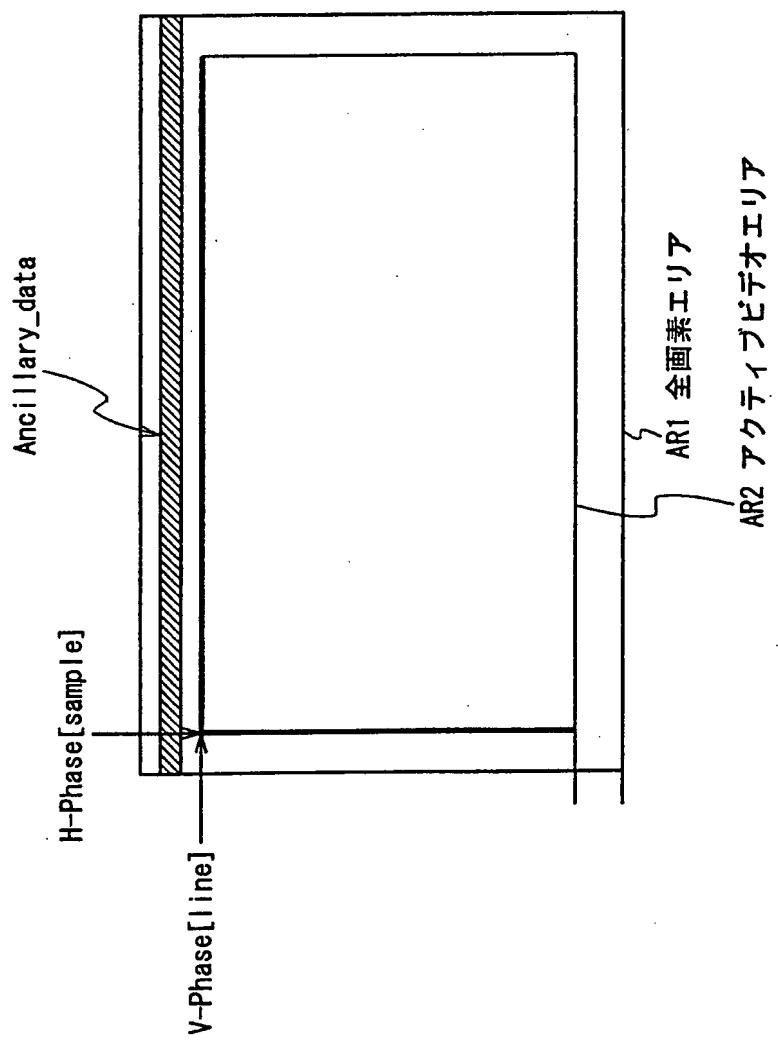


図 8

図9 A

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
Frame No	1	2	3	4	5	6	7	8	9	10	11	12	13
Picture type	I	B	B	P	B	B	P	B	B	I	B	B	P
Repeat_first_field	1	0	1	0	1	0	1	0	1	0	1	0	1
Top_field_first	1	0	0	1	1	0	0	1	1	0	0	1	1
Frame Structure													

図9 B

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
Frame No	1	2	3	4	5	6	7	8	9	10	11	12	13
Picture type	I	B	B	P	B	B	P	B	B	I	B	B	P
Repeat_first_field	1	0	1	0	1	0	1	0	1	0	1	0	1
Top_field_first	1	0	0	1	1	0	0	1	1	0	0	1	1
Frame Structure													
PTS_counter	0	3	5	8	10	13	15	18	20	23	25	28	30

図9 C

	F1	F4	F2	F3	F7	F5	F6	F10	F8	F9	F13	F11	F12
Frame No	1	4	2	3	7	5	6	10	8	9	13	11	12
Picture type	I	P	B	B	P	B	B	I	B	B	P	B	B
Repeat_first_field	1	0	0	1	1	1	0	0	0	1	1	1	0
Top_field_first	1	1	0	0	0	1	0	0	1	1	1	0	1
Frame Structure													
DTS_counter	125	0	3	5	8	10	13	15	18	20	23	25	28

	No. of bits	Mnemonic
video_sequence()		
next_start_code()		
sequence_header()		
sequence_extension()		
do{		
extension_and_user_data(0)		
do{		
if(nextbits() == group_start_code){		
group_of_pictures_header()		
extension_and_user_data(1)		
}		
picture_header()		
picture_coding_extension()		
extension_and_user_data(2)		
picture_data()		
}while((nextbits() == picture_start_code) !=		
(nextbits() == group_start_code))		
if(nextbits() != sequence_end_code){		
sequence_header()		
sequence_extension()		
}		
}while(nextbits() != sequence_end_code)		
sequence_end_code	32	bslbf
)		

図10

sequence_header()	No. of bits	Mnemonic
sequence_header_code	32	bslbf
horizontal_size_value	12	uimsbf
vertical_size_value	12	uimsbf
aspect_ratio_information	4	uimsbf
frame_rate_code	4	uimsbf
bit_rate_value	18	uimsbf
marker_bit	1	"1"
vbv_buffer_size_value	10	uimsbf
constrained_parameters_flag	1	
load_intra_quantiser_matrix	1	
if(load_intra_quantiser_matrix)		
intra_quantiser_matrix [64]	8*64	uimsbf
load_non_intra_quantiser_matrix	1	
if(load_non_intra_quantiser_matrix)		
non_intra_quantiser_matrix [64]	8*64	uimsbf
next_start_code()		
}		

図 1 1

picture_data()	No. of bits	Mnemonic
do{		
slice()		
}while(nextbits()--slice_start_code)		
next_start_code()		
}		

図 2 5

sequence_extention(){	No. of bits	Mnemonic
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
profile_and_level_indication	8	uimsbf
progressive_sequence	1	uimsbf
chroma_format	2	ulmsbf
horizontal_size_extension	2	uimsbf
vertical_size_extension	2	ulmsbf
bit_rate_extension	12	uimsbf
marker_bit	1	bslbf
vbv_buffer_size_extension	8	ulmsbf
low_delay	1	uimsbf
frame_rate_extension_n	2	unimsbf
frame_rate_extension_d	5	ulmsbf
next_start_code()		
}		

図 1 2

extension_and_user_data(i){	No. of bits	Mnemonic
while((i==1)lts()&&(nextbits()==extension_start_code)){		
(nextbits()==user_data_start_code)){		
if(nextbits()==extension_start_code)		
extension_data(i)		
if(nextbits()==user_data_start_code)		
user_data()		
}		
}		

図 1 3

Syntax	Bits	Mnemonic
user_data0 {		
user_data_start_code	32	
MPEG_ES_Editing_information(i) {		
if(i==0) {		
while(nextbits()!= "0000 0000 0000 0000 0000 0001") {		
if(nextbits()== "V-phase")		
V-phase()		
Else if(nextbits()== "H-phase")		
H-phase()		
{		
{		
Else if(i==2) {		
while(nextbits()!= "0000 0000 0000 0000 0000 0001") {		
if((nextbits()== "Time code1") (nextbits()== "Time code2"))		
Time_code()		
Else if(nextbits()== "Picture Order")		
Picture_order()		
{		
while(nextbits()!= "0000 0000 0000 0000 0000 0001") {		
if(nextbits()== "Ancillary_data")		
Ancillary_data()		
Else if(nextbits()== "History data")		
History data()		
{		
{		
if(nextbits()== "User data")		
User_data()		
Next_start_code()		
}		

图14

Data_ID	Data_type
00	FORBIDDEN
01	V-Phase
02	H-Phase
03	Time code 1
04	Time code 2
05	Picture Order
06	Video Index
07	Ancillary data
08	History data
...	...
80	Control flags
...	...
FF	User data

図 15

Syntax	Bits	Mnemonic
V-Phase(){		
Data_ID	8	bslbf
V-Phase	16	uimsbf
}		

図 16

Syntax	Bits	Mnemonic
H-Phase(){		
Data_ID	8	bslbf
H-Phase	8	uimsbf
}		

図17

Syntax	Bits	Mnemonic
Picture_order(){		
Data_ID	8	bslbf
DTS_presence	1	bslbf
PTS_counter	7	uimsbf
If(DTS_presence== "1"){		
Marker_bits	1	bslbf
DTS_counter	7	uimsbf
}		
}		

図20

Syntax	Bits	Mnemonic
Time_code(){		
Data_ID	8	bslbf
Time_code [63..48]	16	uimsbf
Marker_bit	1	bslbf
Time_code [47..32]	16	uimsbf
Marker_bit	1	bslbf
Time_code [31..16]	16	uimsbf
Marker_bit	1	bslbf
Time_code [15..0]	16	uimsbf
Marker_bit	1	bslbf
Reserved_bits	4	bslbf
}		

図 1 8

Syntax	Bits	Mnemonic
Time_code [63..0]{		
Color frame flag	1	bslbf
Drop frame flag (NTSC)/unused (PAL)	1	bslbf
TV frame tens	2	uimsbf
TV frame units	4	uimsbf
Field phase (NTSC)/binary group flag 0 (PAL)	1	bslbf
TV seconds tens	3	uimsbf
TV seconds units	4	uimsbf
Binary group flag 0 (NTSC)/binary group flag 2 (PAL)	1	bslbf
TV minutes tens	3	uimsbf
TV minutes units	4	uimsbf
Binary group flag 2 (NTSC)/field phase (PAL)	1	bslbf
Binary group flag 1 (NTSC)/binary group flag 1 (PAL)	1	bslbf
TV hours tens	2	uimsbf
TV hours units	4	uimsbf
2nd binary group	4	uimsbf
1st binary group	4	uimsbf
4th binary group	4	uimsbf
3rd binary group	4	uimsbf
6th binary group	4	uimsbf
5th binary group	4	uimsbf
8th binary group	4	uimsbf
7th binary group	4	uimsbf
}	4	uimsbf

図 1 9

Syntax	Bits	Mnemonic
Ancillary_data(){		
Data_ID	8	bslbf
Field_ID	2	bslbf
Line_number	14	uimsbf
Ancillary_data_length	16	uimsbf
Marker_bits	1	bslbf
For(j=0; j<Ancillary_data_length; j++){		
Ancillary_data_payload	22	uimsbf
Marker_bits	1	bslbf
}		
While(!bytealigned())		
Zero_bit	1	“0”
}		

図 2 1

	No. of bits	Mnemonic
group_of_picture_header(){		
group_start_code	32	bslbf
time_code	25	bslbf
closed_gop	1	uimsbf
broken_link	1	uimsbf
next_start_code()		
}		

図 2 2

picture_header(){	No. of bits	Mnemonic
picture_start_code	32	bslbf
temporal_reference	10	uimsbf
picture_coding_type	3	uimsbf
vbv_delay	16	uimsbf
if(picture_coding_type==2 picture_coding_type==3){		
full_pel_forward_vector	1	
forward_f_code	3	uimsbf
}		
if(picture_coding_type==3){		
full_pel_backward_vector	1	
backward_f_code	3	uimsbf
}		
while(nextbits()== "1"){		
extra_bit_picture/*with the value "1" */	1	uimsbf
extra_information_picture	8	
}		
extra_bit_picture/*with the value "0" */	1	uimsbf
next_start_code()		
}		

図23

picture_coding_extension(){	No. of bits	Mnemonic
extension_start_code	32	bslbf
extension_start_code_identifier	4	uimsbf
f_code [0][0]/*forward horizontal*/	4	uimsbf
f_code [0][1]/*forward vertical*/	4	uimsbf
f_code [1][0]/*backward horizontal*/	4	uimsbf
f_code [1][1]/*backward vertical*/	4	uimsbf
intra_dc_precision	2	uimsbf
picture_structure	2	uimsbf
top_field_first	1	uimsbf
frame_pred_frame_dct	1	uimsbf
concealment_motion_vectors	1	uimsbf
q_scale_type	1	uimsbf
intra_vlc_format	1	uimsbf
alternate_scan	1	uimsbf
repeat_first_field	1	uimsbf
chroma_420_type	1	uimsbf
progressive_frame	1	uimsbf
composite_display_flag	1	uimsbf
if(composite_display_flag){		
v_axis	1	uimsbf
field_sequence	3	uimsbf
sub_carrier	1	uimsbf
burst_amplitude	7	uimsbf
sub_carrier_phase	8	uimsbf
}		
next_start_code()		
}		

図24

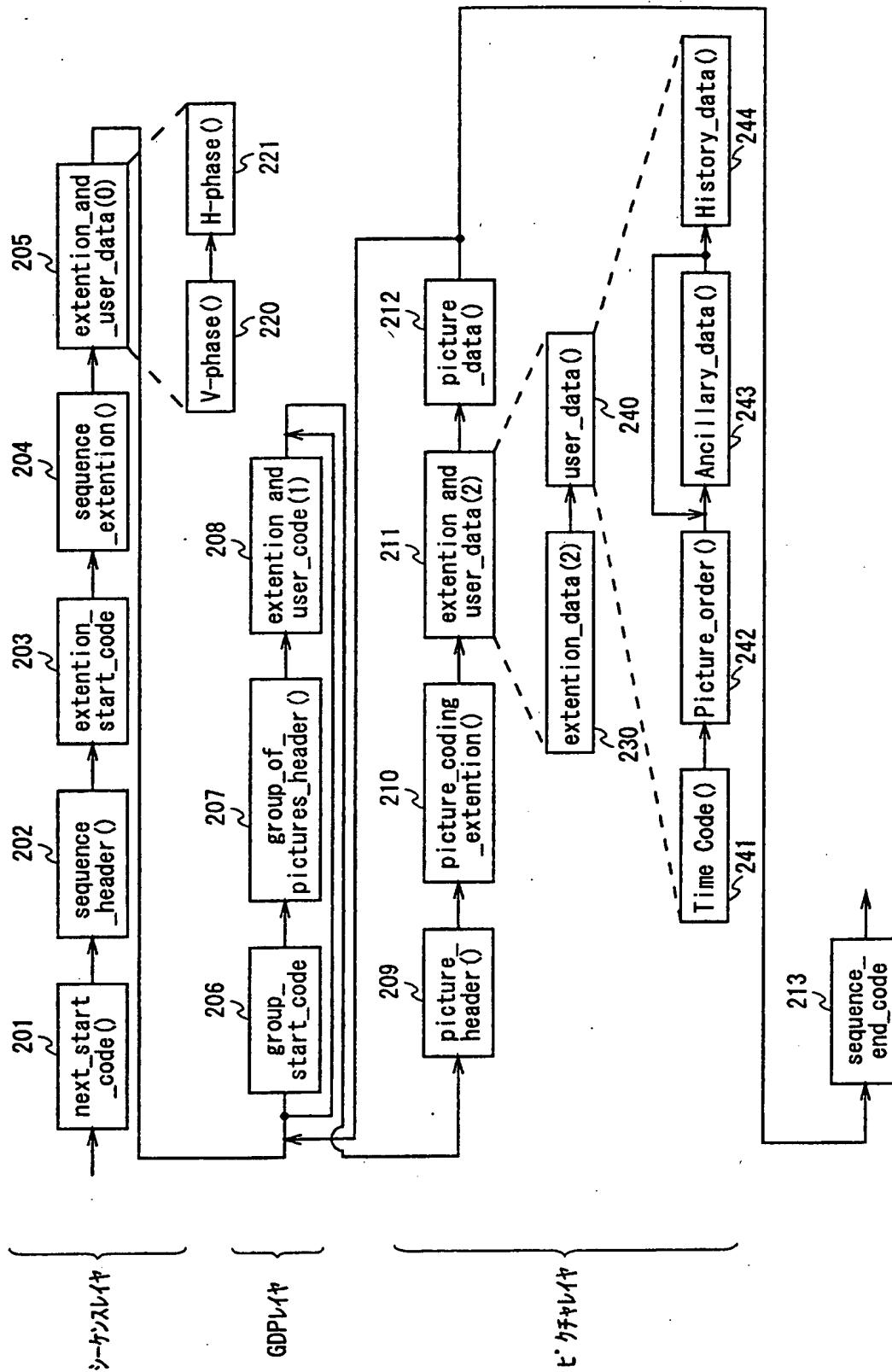


図 26

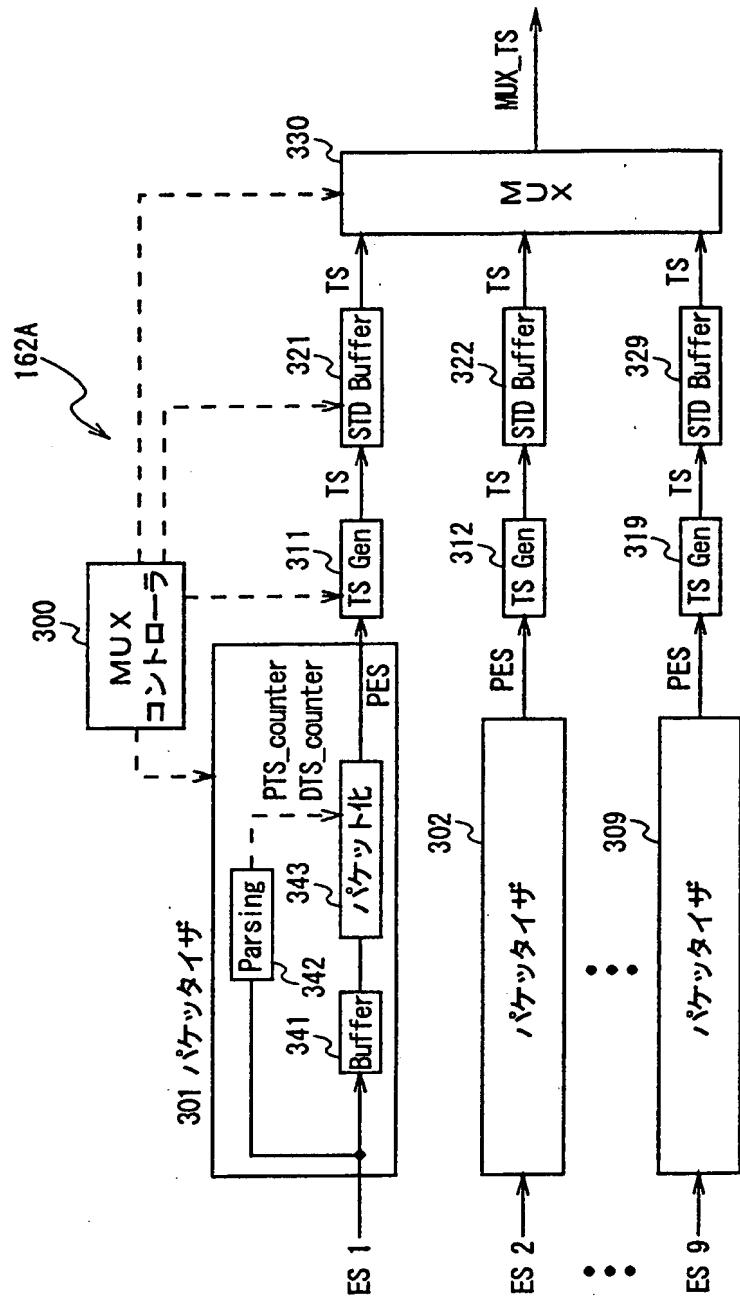
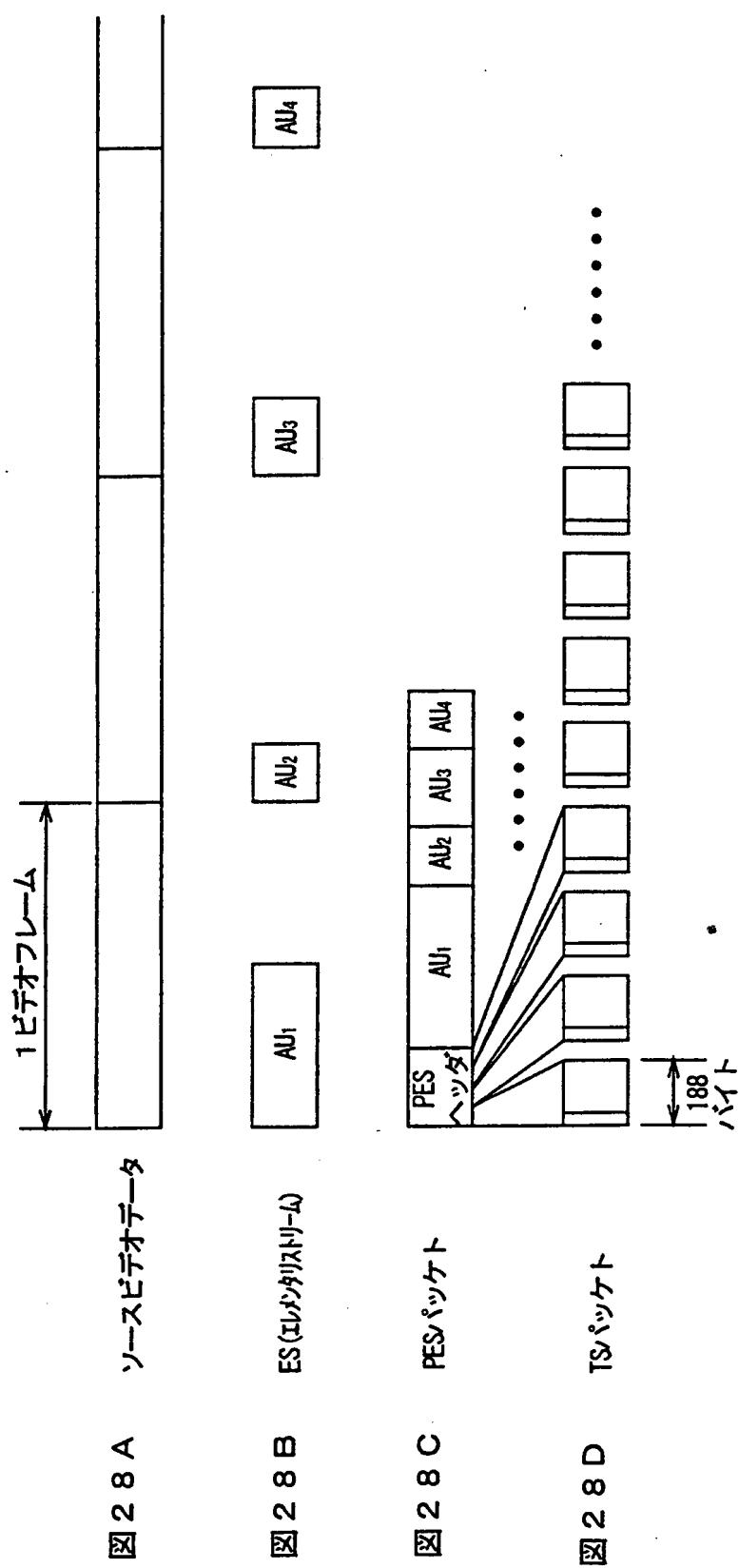


図27



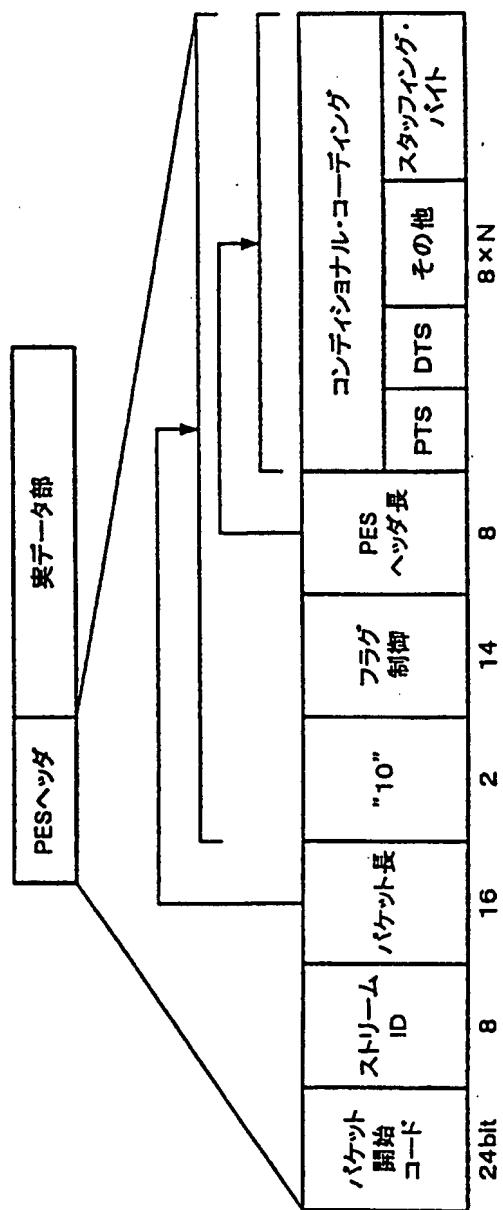


図 29

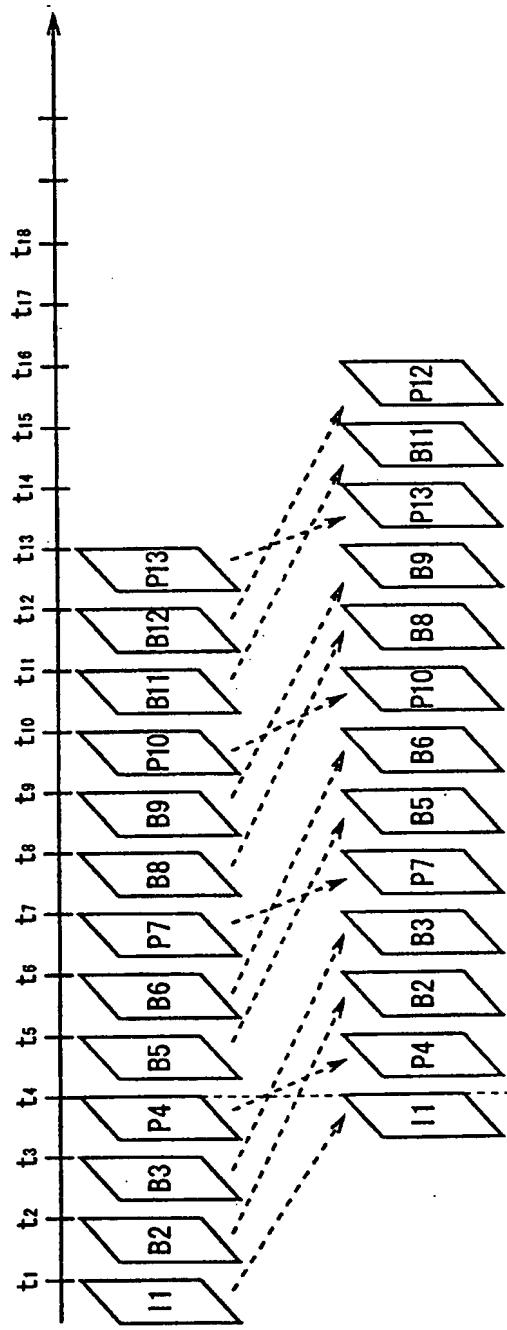
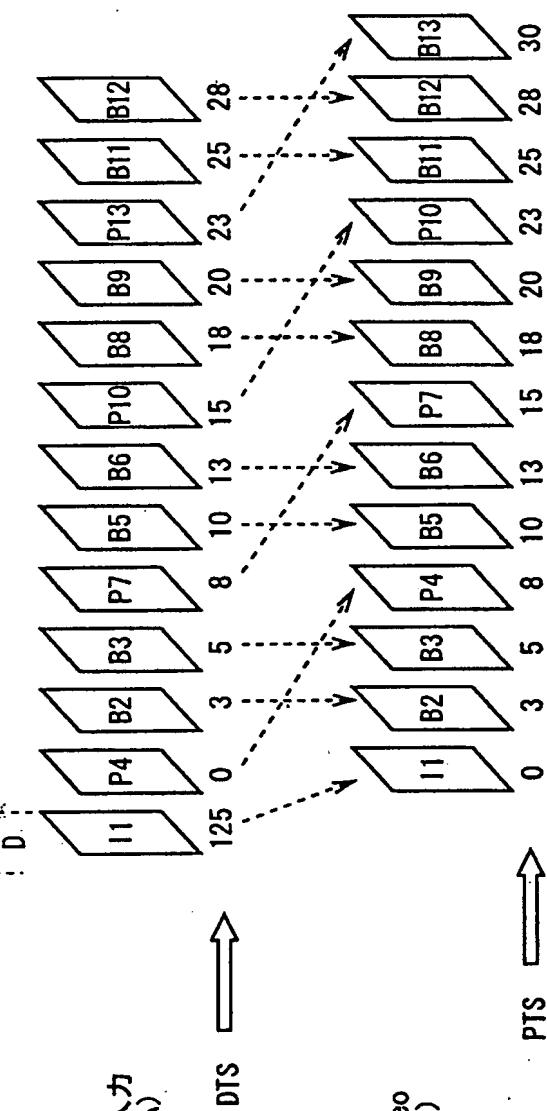
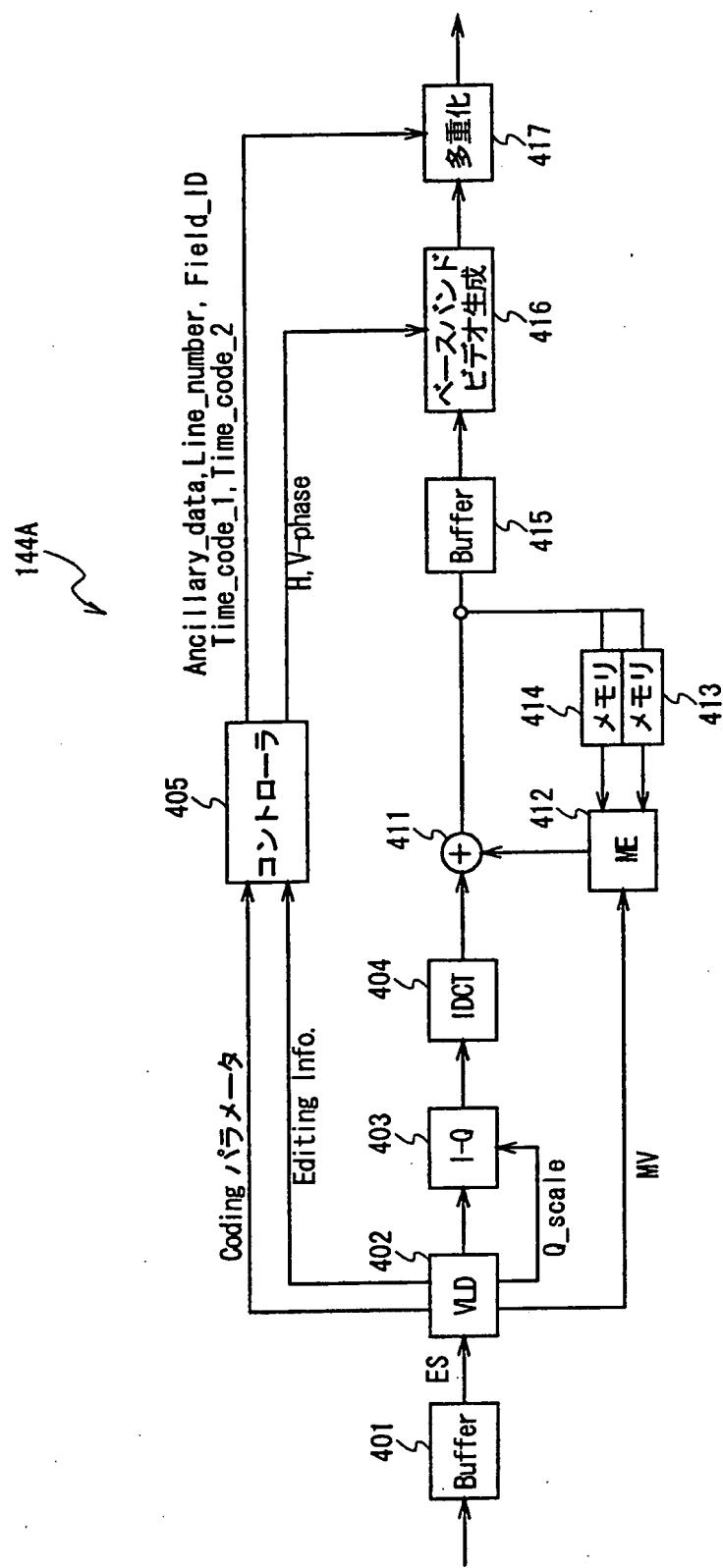


图 30 B ES

图 30 C PES/TS
(データ入力されるストリーム)
DTS图 30 D Decoded Video
(表示ビデオ)
PTS



1
3
EX

符 号 の 説 明

1 ……ビデオプロセッサ、2 ……MPEGエンコーダ、3 ……MPEGデコーダ、5 ……3-2プルダウン処理回路、6 ……逆3-2プルダウン処理回路、104 ……コントローラ、105 ……動き検出回路、112 ……DCT回路、113 ……量子化回路、119 ……動き補償回路、125 ……可変長符号化回路、126 ……送信バッファ、142A ……MPEGエンコーダ、300 ……多重化コントローラ、301、302、309 ……パケットタイザ、330 ……マルチプレクサ。

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP00/00628

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁷ H04N7/08, H04N7/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ H04N7/025-7/088, H04N7/24-7/68Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2000
Kokai Jitsuyo Shinan Koho 1971-2000 Jitsuyo Shinan Toroku Koho 1996-2000Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
JOIS (JICST, SDI)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP, 740474, A2 (Oki Electric Industry Co., Ltd.), 12 November, 1996 (12.11.96), Full text; Figs. 1 to 3	1,11-16,18, 20,25,30,35, 38-42,45-48
Y	Full text; Figs. 1 to 3 & JP, 8-298649, A	2-10,17,19, 21-24,26-29, 31-34,36,37,43 ,44
X	JP, 6-86250, A (NEC Corporation), 25 March, 1994 (25.03.94), Full text; Figs. 1 to 8	1,7,11-16, 18,20
Y	Full text; Figs. 1 to 8 (Family: none)	2-6,8-10,17,19 ,21-24
Y	EP, 854654, A2 (Nextlevel Systems, Inc.), 21 August, 1998 (21.08.98), Full text; Figs. 1 to 5	2-6,8,17,19,21 -24,26-28,31-3 4,36,37,43,44

 Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family
---	--

Date of the actual completion of the international search 02 May, 2000 (02.05.00)	Date of mailing of the international search report 16 May, 2000 (16.05.00)
--	---

Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer
Facsimile No.	Telephone No.

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP00/00628

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	& JP, 10-224751, A & US, 5844615, A	
Y	JP, 10-150630, A (Victor Company of Japan, Limited), 02 June, 1998 (02.06.98), Full text; Figs. 1 to 16	2,8,17,23, 26,31-34,36
	(Family: none)	
Y	JP, 9-46705, A (Matsushita Electric Ind. Co., Ltd.), 14 February, 1997 (14.02.97), Full text; Figs. 1 to 15 (Family: none)	9,29
Y	JP, 9-261195, A (Sony Corporation), 03 October, 1997 (03.10.97), Full text; Figs. 1 to 21 (Family: none)	2,29
Y	JP, 7-298212, A (Sony Corporation), 10 November, 1995 (10.11.95), Full text; Figs. 1 to 9 (Family: none)	10

A. 発明の属する分野の分類 (国際特許分類 (IPC))
Int. Cl' H04N7/08, H04N7/24

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))
Int. Cl' H04N7/025-7/088, H04N7/24-7/68

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2000年
日本国登録実用新案公報	1994-2000年
日本国実用新案登録公報	1996-2000年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)
JOIS (JICSTファイル、SDI)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X	E P, 740474, A2 (沖電気工業株式会社) 17. 11月. 1996 (12. 11. 96) 全文, 第1-3図	1, 11-16, 18, 20, 25, 30, 35, 38-42, 45-48
Y	全文, 第1-3図 & J P, 8-298649, A	2-10, 17, 19, 21-24, 26-29, 31-34, 36, 37, 43, 44

C欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示す
もの
「E」国際出願日前の出願または特許であるが、国際出願日
以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行
日若しくは他の特別な理由を確立するために引用する
文献 (理由を付す)
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって
出願と矛盾するものではなく、発明の原理又は理
論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明
の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以
上の文献との、当業者にとって自明である組合せに
よって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日

02. 05. 00

国際調査報告の発送日

16.05.00

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)
郵便番号 100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

梅本 達雄

5P 9648

電話番号 03-3581-1101 内線 3581

C (続き) 関連すると認められる文献		関連する請求の範囲の番号
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	
X	JP, 6-86250, A (日本電気株式会社) 25. 3月. 1994 (25. 03. 94) 全文, 第1-8図	1, 7, 11-16, 18, 20
Y	全文, 第1-8図 (ファミリーなし)	2-6, 8-10, 17, 19, 21-24
Y	EP, 854654, A2 (エストレーブル・システムズ・インコーポ・レイテッド) 21. 8月. 1998 (21. 08. 98) 全文, 第1-5図	2-6, 8, 17, 19, 21-24, 26-28, 31-34, 36, 37, 43, 44
Y	& JP, 10-224751, A & US, 5844615, A JP, 10-150630, A (日本ビクター株式会社) 2. 6月. 1998 (02. 06. 98) 全文, 第1-16図 (ファミリーなし)	2, 8, 17, 23, 26, 31-34, 36
Y	JP, 9-46705, A (松下電器産業株式会社) 14. 2月. 1997 (14. 02. 97) 全文, 第1-15図 (ファミリーなし)	9, 29
Y	JP, 9-261195, A (ソニー株式会社) 3. 10月. 1997 (03. 10. 97) 全文, 第1-21図 (ファミリーなし)	2, 29
Y	JP, 7-298212, A (ソニー株式会社) 10. 11月. 1995 (10. 11. 95) 全文, 第1-9図 (ファミリーなし)	10